Informatik-Grundlagen

- Klärung „Informatik“
- Geschichte der Informatik
- Algorithmus
- Software, Programm, Programmentwicklung
- Von-Neumann-Rechner

Was ist Informatik?

Klärung Informatik

- **Der Begriff "Informatik"**
  - ein ***Kunstwort*** , das zu Beginn der 60er Jahre zur Bezeichnung einer sich neu entwickelnden Disziplin geschaffen wurde. Es setzt sich aus Bestandteilen der beiden Worte ***Information*** und ***Mathematik*** zusammen. Darin kommt zum Ausdruck, dass Informatik die Wissenschaft von der Informationsverarbeitung ist, und eine große Nähe zur Mathematik hat. Heute wird der Begriff z.T. sehr anwendungsnah gebraucht und Synonym zur ‘Informationstechnik‘ verwendet.
- **Informatik ist die Wissenschaft**
  - von der ***systematischen*** Verarbeitung von Informationen, besonders der automatischen Verarbeitung mit Hilfe von ***Computern*** (vgl. DUDEN Informatik, 1993)
- **Informatik versteht sich als Wissenschaft**
  - der Analyse, Konzeption und Realisierung von Systemen, die aus miteinander und mit ihrer Umwelt kommunizierenden Akteuren bestehen (vgl. Studienführer Informatik, RWTH Aachen, 1998)

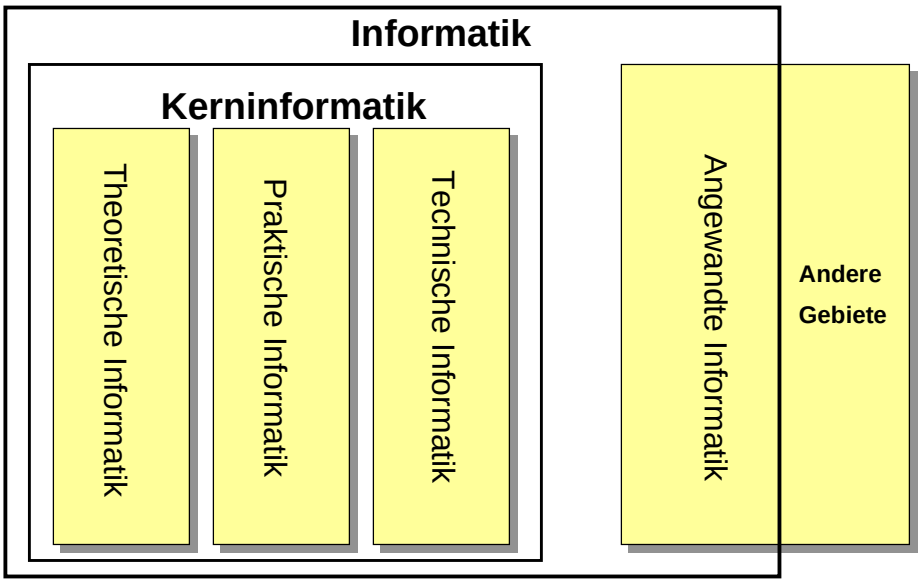
Hauptaufgaben der Informatik

Klärung Informatik

- **Hauptaufgabe der Informatik ist die Entwicklung**
  - ***formaler, maschinell ausführbarer Verfahren*** zur Lösung von Informationsverarbeitungsproblemen, die häufig als Teilprobleme komplexer Kommunikations- oder Organisationsprobleme auftreten.
- **Die Forderung der Durchführbarkeit mittels einer Maschine (i.a. eines Digitalrechners) bedingt,**
  - dass die zu verarbeitenden Informationen als ***maschinell verarbeitbare Daten*** dargestellt werden, und
  - dass die Lösungsverfahren bis ***ins Detail*** formal beschrieben werden.

Gebiete der Informatik - 1

Klärung Informatik



Klärung Informatik

Gebiete der Informatik - 2

- **Theoretische Informatik**
  - ***Formale Modelle*** zur Beschreibung und Untersuchung von Algorithmen, Computern, etc.
  - Teilgebiete: Formale Sprachen, Automatentheorie, Komplexitätstheorie etc.
- **Technische Informatik**
  - Funktioneller ***Aufbau von Computern***, Entwurf und Entwicklung von Rechnern, Geräten und Schaltungen
  - Teilgebiete: Rechnerarchitektur, VLSI-Entwurf etc.
- **Praktische Informatik**
  - ***Prinzipien und Techniken*** der Fundierung und Realisierung in Software (großer Programmsysteme!)
  - Teilgebiete: Softwaretechnik, Informationssysteme, Compilerbau, Betriebssysteme, Künstliche Intelligenz, Kommunikation/verteilte Systeme, Parallele Systeme, etc.

Klärung Informatik

Gebiete der Informatik - 3

- **Angewandte Informatik**
  - ***Anwendung*** der Methoden der ***Kerninformatik*** in anderen Wissenschaften
    - Entwicklung ***spezieller*** Verfahren und Darstellungstechniken
  - Bindestrich-Informatik:
    - Wirtschaftsinformatik, Medizin-Informatik, Bioinformatik, Rechts-Informatik
  - Grenzen zwischen ***Praktischer Informatik*** und ***Angewandter Informatik*** sind zum Teil fließend.
- **Das, was man unter Informatik versteht, kann man in solchen Definitionen jedoch nicht *endgültig* fassen. Schließlich ändert sich die Beschreibung der Definitionen einer Wissenschaft, wie in anderen Fällen auch, über die Zeit.**

Geschichte der Informatik

Geschichte - 1

- **Altertum–Mittelalter:**
  - Verwendung des ***Abakus*** (Brett mit verschiebbaren Kugeln) als Hilfsmittel für die vier Grundrechenarten.
- **9. JH.:**
  - Der arabische Mathematiker und Astronom ***Ibn Musa Al-Chwarismi*** schreibt das Lehrbuch "Kitab al jabr w' almuqabala" ("Regeln der Wiedereinsetzung und Reduktion"). Das Wort "***Algorithmus***" geht auf seinen Namen zurück.
- **1547: Adam Riese (1492–1559)**
  - veröffentlicht ein Rechenbuch, in dem er die ***Rechengesetze*** des aus Indien stammenden ***Dezimalsystems*** (5. Jh.n. Chr.) beschreibt. Im 17. Jahrhundert setzt sich das Dezimalsystem in Europa durch.
- **Wilhelm Schickard (1592–1635)**
  - konstruiert für seinen Freund Kepler (1571–1630) eine ***Maschine***, die addieren, subtrahieren, multiplizieren und dividieren kann. Sie bleibt unbeachtet.
- **1641: Blaise Pascal (1623–1662)**
  - konstruiert eine Maschine, mit der man ***sechsstellige Zahlen*** addieren kann.

Geschichte der Informatik

Geschichte - 2

- **1674: Gottfried Wilhelm Leibniz (1646–1716)**
  - konstruiert eine ***Rechenmaschine*** mit Staffelwalzen für die vier ***Grundrechenarten***. In diesem Zusammenhang befasst er sich auch mit der binären Darstellung von Zahlen.
- **1774: Philipp Matthäus Hahn (1739–1790)**
  - entwickelte eine mechanische Rechenmaschine, die ***erstmals zuverlässig*** arbeitet.
- **Ab 1818:**
  - Rechenmaschinen nach dem Vorbild der Leibnizschen Maschine werden serienmäßig hergestellt und dabei ständig weiterentwickelt.
- **1838: Charles Babbage (1792–1871)**
  - plant eine Maschine, die "***Analytical Engine***", bei der die Reihenfolge der einzelnen Rechenoperationen durch nacheinander eingegebene ***Lochkarten*** gesteuert wird.
- **1886: Hermann Hollerith (1860–1929)**
  - entwickelt in den USA elektrisch arbeitende ***Zählmaschinen für Lochkarten***, mit denen die statistischen Auswertungen der Volkszählungen vorgenommen werden.

Geschichte - 3

Geschichte der Informatik

- **1934: Konrad Zuse (1910–1995)**
  - beginnt mit der Planung einer ***programmgesteuerten Rechenmaschine***. Sie verwendet das binäre Zahlensystem.
- **1937: Die mechanische Anlage *Z 1* von Zuse ist fertig.**
- **1941: Die elektromechanische Anlage Z 3 von Zuse ist fertig.**
  - Dies ist der ***erste funktionsfähige programmgesteuerte Rechenautomat***. Das Programm wurde mit ***Lochstreifen*** eingegeben. Die Anlage verfügt über 2000 Relais und eine Speicherkapazität von 64 Worten a 22 Bit. Multiplikationszeit: etwa 3 s.
- **1944: Howard H. Aiken (1900–1973)**
  - erstellt in Zusammenarbeit mit der Harvard-University und der Firma IBM die teilweise programmgesteuerte Rechenanlage MARK I. Additionszeit 1/3 s, Multiplikationszeit: 6 s.
- **1946: J. P. Eckert und J. W. Mauchly**
  - stellen die ENIAC (Electronic Numerical Integrator and Automatic Calculator) fertig. Dies ist der erste ***voll elektronische Rechner*** (18.000 Elektronenröhren). Multiplikationszeit: 3 ms.

Geschichte - 4

Geschichte der Informatik

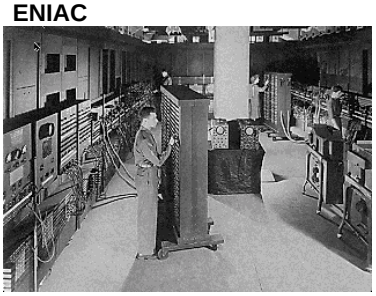
- **1946–1952:**
  - Auf der Grundlage der Ideen **John v. Neumanns** (1903–1957) (Einzelprozessor, Programm und Daten im gleichen Speicher; Von-Neumann-Rechner) und seiner Kollegen am Institute of Advanced Study at Princeton (H.H.Goldstine, A.W.Burks) werden weitere Computer in Universitätslabors entwickelt (“Pionierzeit”).
- **1949: M.V. Wilkes (University of Manchester)**
  - stellt mit der EDSAC (Electronic Delay Storage Automatic Calculator) den **ersten universellen Digitalrechner** (gespeichertes Programm) fertig.
- **Ab 1950:**
  - **Industrielle** Rechnerentwicklung und -produktion.

Historische Rechner

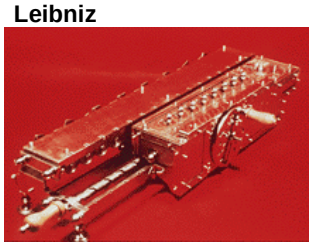
Geschichte der Informatik

Schickard

ENIAC



Leibniz



MARK1

Informelle Definition

Algorithmus

- **Ein Algorithmus ist ein Verfahren, welches**
  - in einem ***endlichen*** Text niedergelegt werden muss
  - ***effektiv*** ausführbar ist,
  - Elementaroperationen enthält, die durch die jeweilige Situation ***eindeutig*** bestimmt sind
  - ***Ein- und Ausgabe*** ermöglicht
  - durch eine (mechanisch oder elektronisch arbeitende) ***Maschine ausgeführt*** werden kann.
- **Anzahl und Ausführungszeit der Elementaroperationen sind beschränkt.**
- **Ein Algorithmus (Programm) wird durch eine Maschine schrittweise ausgeführt**
  - Die ausführende Instanz muss die Vorschrift ***interpretieren*** und ***korrekt*** ausführen.
  - Ein Algorithmus ***terminiert***, wenn er nach endlich vielen Schritten abbricht.

Algorithmus

Euklidischer Algorithmus - 1

Euklidischer Algorithmus

- Problem:
  - Man bestimme zu je zwei natürlichen Zahlen n und m den größten gemeinsamen Teiler ggt(n,m)

Algorithmus

Wiederhole:

❶ WENN n < m ist, DANN vertausche man n und m

❷ WENN m = 0 ist, DANN ist n der ggt(n,m) und man beende den Algorithmus

❸ WENN m≠ 0 ist, DANN bilde man den Rest r, der bei der Division von n durch m bleibt, dann ersetze man n durch m und m durch r und beginne von vorn.

Algorithmus

Euklidischer Algorithmus - 2

n = 6
m = 10

Ablaufverfolgung (Trace)

für Überprüfung von Programmen

	n	m
Erster Durchlauf	**6**	**10**
da 6 < 10, so setzt man n = 10 und m = 6	10	6
da 6 ≠ 0 ist, gehen zu Schritt 3	10	6
r = 4. Man erhält also n = 6 und m = 4	6	4
Zweiter Durchlauf		
Da 6 ≥ 4 ist, gehe zu Schritt 2	6	4
Da 4 ≠ 0 ist, gehe zu Schritt 3	6	4
r = 2. Man erhält n = 4 und m = 2	4	2
Dritter Durchlauf		
Da 4 ≥ 2 ist, gehe zu Schritt 2	4	2
Da 2 ≠ 0 ist, gehe zu Schritt 3	4	2
r = 0. Man erhält n = 2 und m = 0	2	0
Abbruch		
Da 2 ≥ 0 ist, gehe zu Schritt 2	2	0
Da m = 0 ist, Programmende	2	0

Ergebnis: ggt(10, 6) =2

Algorithmus

Eigenschaften - 1

- **Abstraktion**
  - ein Algorithmus löst i. a. eine ***Klasse von Problemstellungen*** (z.B. Suchen eines Musters in einer Zeichenkette)
- **Finitheit**
  - statisch finit: ein Algorithmus besitzt eine ***endliche Länge***
  - dynamisch finit: während der Abarbeitung darf nur ***endlich viel Speicherplatz*** belegt werden
- **Terminierung**
  - terminierend: nach ***endlich vielen Schritten*** liegt ein Resultat vor
  - sonst ***nicht-terminierend*** (z.B. Steuerungsalgorithmen)
- **Determinismus**
  - deterministisch: zu jedem Zeitpunkt besteht ***höchstens eine*** Möglichkeit der Fortsetzung
  - nicht-deterministisch: an mindestens einer Stelle gibt es eine ***Wahlmöglichkeit*** für die Fortsetzung

Algorithmus

Eigenschaften - 2

- **Determiniertheit**
  - determiniert: bei ***gleichen Eingaben*** und Startbedingungen wird das ***gleiche Ergebnis*** erzielt
  - nicht-determiniert: es werden ***unterschiedliche Ergebnisse*** erzielt (z.B. Anwendung von heuristischen Methoden, syst. Probieren)
- **Bemerkung**
  - ein terminierender, deterministischer Algorithmus ist immer determiniert
  - ein terminierender, nicht-deterministischer Algorithmus kann determiniert oder nicht-determiniert sein.

Algorithmus

Terminierung - Nichtterminierung

- **Sei**
  - power : $N^+ \times N \rightarrow N$

$$\text{power}(a, b) = \begin{cases} 1 & \text{falls } b = 0 \\ a * \text{power}(a, b-1) & \text{sonst} \end{cases}$$

Dieser Algorithmus ***terminiert*** im Definitionsbereich.

  - Wir weiten den Definitionsbereich aus:

$$\text{power2} : Z \times Z \rightarrow Z$$

$$\text{power2}(a, b) = \begin{cases} 0 & \text{falls } a = 0 \text{ und } b > 0 \\ 1 & \text{falls } a \neq 0 \text{ und } b = 0 \\ a * \text{power2}(a, b-1) & \text{sonst} \end{cases}$$

Dieser Algorithmus ***terminiert nicht*** für power2 (0,0) und power2 (a,b) mit $b < 0$ und $a \neq 0$)

Algorithmus

Anmerkung zur Nichtterminierung

Während wir die Nichtterminierung bei der ***manuellen Auswertung*** leicht feststellen können, fällt dies auf dem Rechner sehr viel schwerer. Solange der Rechner ***vor sich hin*** rechnet, können wir den Unterschied zwischen einem ***nicht-terminierenden*** und einem ***sehr langwierigen Algorithmus*** nicht feststellen.

Algorithmus

Fragen

- **Wie kann man aus einer Lösungsidee einen Algorithmus konstruieren?**
  - "schrittweise Programmentwicklung"
- **Wie kann man Algorithmen darstellen?**
  - "Flussdiagramme", Programme
- **Wie beweist man, dass ein Algorithmus tatsächlich das tut, was er tun soll?**
  - Verifikation: partielle Korrektheit
  - Termination
- **Wie "gut" ist ein Algorithmus?**
  - Speicherverbrauch, benötigte Zeit
  - Aufwandsabschätzungen

Algorithmus

Typische Problemklassen

- **Sortieralgorithmen**
  - Ordnen von Elementen
- **Suchalgorithmen**
  - Auffinden von Elementen
- **Algorithmen zur Verarbeitung von Zeichenfolgen**
  - Mustererkennung, Verschlüsselung, Komprimierung
- **Geometrische Algorithmen**
  - z.B. Schnittmenge geometrischer Objekte
- **Algorithmen für Graphen**
  - Suchen im Graph, kürzester Weg
- **Mathematische Algorithmen**
  - Rechnen mit Polynomen und Matrizen
- **Viele Anwendungsalgorithmen:** Kontrolle, Steuerung, Simulation

Standard-Algorithmen der Informatik

Software, Programm, Programmentwicklung

Überblick

- **Neben der Entwicklung der Hardware (Rechner)**
  - wurden seit 1955 ***Programmiersprachen*** entwickelt, um Algorithmen zu formulieren, damit sie von einem Rechner ausgeführt werden können.
- **Rechner "realisieren" Algorithmen,**
  - durch ***schrittweise Abarbeitung*** von ***Programmen***, die in einer Programmiersprache geschrieben sind.
- **Um einen Rechner zu programmieren,**
  - muss die ***Syntax*** und ***Semantik*** der verwendeten Programmiersprache bekannt sein.
- **In diesem Zusammenhang spricht man häufig auch von Software**
  - "***Software***" und "***Programm***" sind aber nicht dasselbe

Software, Programm, Programmentwicklung

Definition: Software

- **1. Definition: Software**
  - Informatik-Duden: Gesamtheit ***aller Programme***, die auf einer Rechenanlage eingesetzt werden können
    - ***Systemsoftware***: Programme die für den korrekten Ablauf einer Rechenanlage notwendig sind
    - ***Anwendungssoftware***: dient zur Lösung von Benutzerproblemen
- **2. Definition: Software**
  - IEEE Standard Glossary of Software Engineering Terminoloy
    - "Computer ***programs***, ***procedures***, and possibly associated ***documentation*** and ***data*** pertaining to the operation of a computer system."

Testfälle, Handbuch, Installations-anweisung etc.

Software, Programm, Programmentwicklung

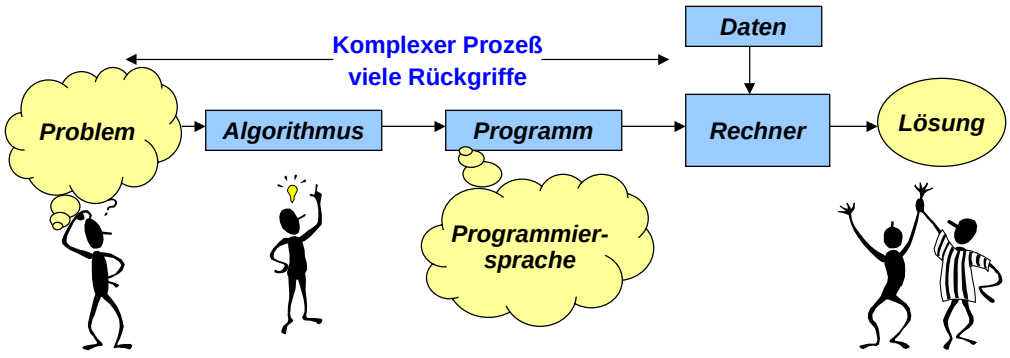
Definition: Programm

- **1. Definition: Programm in einer Programmiersprache**
  - Formulierung eines Algorithmus und der dazugehörigen Datenbereiche in einer Programmiersprache.
  - Ein Programm
    - ist ***exakt*** (formal) definiert
    - nimmt Bezug auf eine bestimmte Darstellung der ***Daten***
    - ist auf einer Rechenanlage ***ausführbar***
- **2. Definition: Programm**
  - IEEE Standard Glossary of Software Engineering Terminoloy
    - "A combination of ***computer instructions*** and ***data definitions*** that enable computer hardware to ***perform*** computational or control functions".

Software, Programm, Programmentwicklung

Programmentwicklung

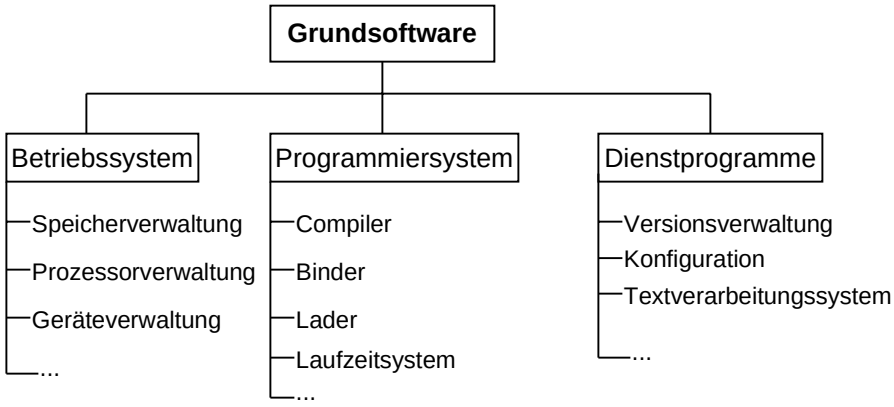
- **Unter dem Begriff Programmieren versteht man**
  - das **Lösen von Problemen** unter Zuhilfenahme eines **Rechners**



- **Programmieren ≠ Software-Entwicklung**
- **Programmieren ist *ein Teil* der Software-Entwicklung**

Software, Programm, Programmentwicklung

Grundsoftware (Systemsoftware)



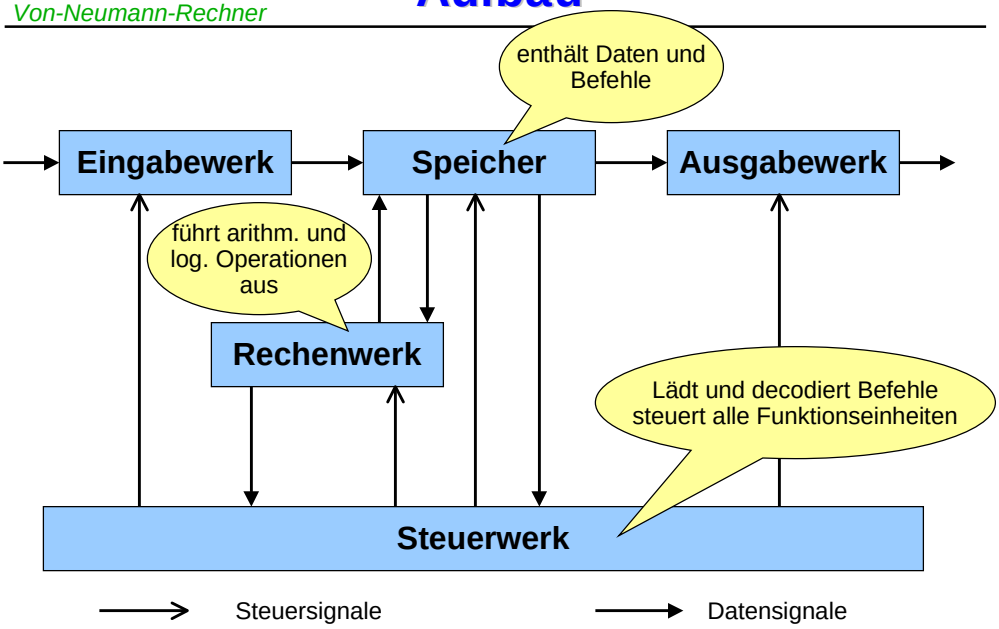
Von-Neumann-Rechner

Überblick

- **Definiert die wesentlichen Elemente eines Universalrechners**
  - Rechner soll nicht für eine ***bestimmte*** Problemklasse konstruiert sein
  - Zur Lösung des Problems muss ein ***Programm*** eingegeben und in den ***Speicher*** abgelegt werden
- **1946 von John von Neumann als Konzept für die EDVAC vorgeschlagen**
- **Fast alle heutigen Rechner basieren darauf und sind Weiterentwicklungen davon**
  - Nicht-von-Neumann-Rechner sind Gegenstand der Forschung
- **Diese Architektur prägt viele Programmiersprachen (imperative Programmiersprachen)**

Aufbau

Von-Neumann-Rechner



Prinzipien - 1

Von-Neumann-Rechner

- Der Rechner besteht aus *Steuerwerk, Rechenwerk, Speicher, Eingabewerk* und Ausgabewerk.
- Die Struktur des von-Neumann-Rechners ist *unabhängig* von den zu bearbeitenden Problemen. Zur Lösung eines Problems muss von außen das *Programm* eingegeben und im Speicher abgelegt werden.
- Programme, Daten, Zwischen- und Endergebnisse werden in *demselben Speicher* abgelegt.
- Der Speicher ist in *gleichgroße Zellen* unterteilt, die fortlaufend durchnumeriert sind. Über die Nummer (*Adresse*) einer Speicherzelle kann deren Inhalt abgerufen oder verändert werden.
- Aufeinanderfolgende Befehle eines Programms werden in *aufeinanderfolgenden* Speicherzellen abgelegt. Das Ansprechen des nächsten Befehls geschieht vom Steuerwerk aus durch Erhöhen der Befehlsadresse um Eins.
- Durch *Sprungbefehle* kann von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abgewichen werden.

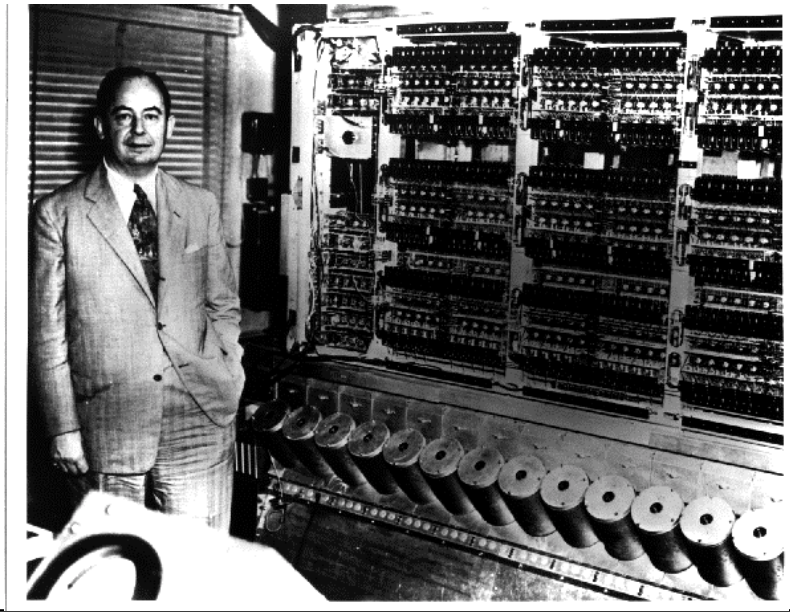
Prinzipien - 2

Von-Neumann-Rechner

- Es gibt zumindest
  - *arithmetische* Befehle wie Addieren, Multiplizieren usw.;
  - *logische* Befehle wie Vergleiche, logisches Nicht, Und, Oder usw.;
  - *Transportbefehle*, z.B. vom Speicher zum Rechenwerk und für die Ein-/Ausgabe;
  - bedingte *Sprünge*.
  - Weitere Befehle wie Schieben, Unterbrechen, Warten usw. kommen hinzu.
- Alle Daten (Befehle, Adressen usw.) werden *binär codiert*. Geeignete Schaltwerke im Steuerwerk und an anderen Stellen sorgen für die richtige Entschlüsselung (*Decodierung*).

Historische Rechner - J. v. Neumann

Von-Neumann-Rechner



Was haben wir gelernt

- **Einordnung der Informatik als Wissenschaft**
  - Aufgabe der Informatik
  - Einteilung der Informatik
- **Wo kommt die Informatik her**
  - Entwicklung der Rechenmaschinen
- **Was versteht man unter einem Algorithmus**
  - Eigenschaften von Algorithmen
- **Was versteht man unter den zentralen Begriffen**
  - Software
  - Programm
  - Programmieren
- **Grober Aufbau eines von-Neumann-Rechners**

Glossar

- **Informatik:**
  - Begriff, Einteilung, Einordnung, Aufgaben, Geschichte
- **Algorithmus:**
  - Definition, Eigenschaften, Klassen von Algorithmen
- **Software**
- **Grundsoftware**
  - Betriebssystem, Programmiersystem, Dienst- und Hilfsprogramme
- **Programm**
- **Programmieren, Programmentwicklung**
- **Von-Neumann-Rechner:**
  - Speicher, Rechenwerk, Steuerwerk. Ein-/Ausgabeeinheit, Befehlsgruppen