Vorlesung 30.10.2023  
Grundlagen 5

### Datentypen Allgemein

* Es gibt eine Vielzahl von Datentypen.
  + Sie bieten viele Grundoperationen
* Datentypen werden genutzt
  + um Datenbereiche einzuschränke
  + elementare Operationen auf den Datentypen anzubieten
  + eine konkrete Semantik einzuführen.
* Datentypen lassen sich in verschiedene Kategorien einsortieren.
  + Es gibt primitive Datentypen.
    - Diese nehmen lediglich einen Wert eines bestimmten Wertebereichs an.
      * ganze Zahlen
      * Festkommazahlen
      * Wahrheitswerte
      * Aufzählungstypen.
  + Und Komplexe Datentypen
    - Zeigertypen und dynamische Datentypen.
      * Ein Zeiger ist eine Referenz auf einen anderen Datentyp, was insbesondere bei der objektorientierten Programmiersprache wichtig ist.
    - Des Weiteren gibt es noch zusammengesetzte Datentypen, die ein Datenkonstrukt sind, das aus einfacheren Datentypen besteht. Ein solches Konstrukt ist zum Beispiel ein Array aus Zeichen oder eine Klasse in einer objektorientierten Programmiersprache.
* Zahlensysteme für numerische und nicht-numerische Informationen sind eine wichtige Grundlage
* diese Informationen lassen sich im Binärsystem darstellen.
  + erleichtert die Implementierung und Prüfung von Informationen.
* Zahl 1 könnte zum Beispiel nur eben diese Zahl 1 darstellen oder für den Wert „wahr“ eines Wahrheitswertes stehen.
* Diese Semantik und die Möglichkeit der Validierung bestehen durch die Einführung und Nutzung von Datentypen.

### Char und Ganzzahlen

* Zu den nicht-numerischen Datentypen gehören
  + Boolean (Wahrheitswerte)
  + Char (Zeichen)
    - String (Zeichenketten).
* Bei den numerischen Werten gibt es verschiedene Datentypen
  + ganzzahlige Werte
    - Ganze Zahlen (Integer) werden in der Informatik in „signed“ und „unsigned“ unterschieden.
      * unsigned Integer ist eine vorzeichenlose (nicht negative) Ganzzahl mit dem Minimalwert null.
    - signed Integer kann dagegen auch negative Ganzzahlen beinhalten.
      * Dafür stehen acht bis 128 Bit zur Verfügung
    - Bezeichnung von ganzzahligen Datentypen ergibt sich aus der Datenbreite.
    - Anstelle der mitunter verwendeten sprechenden Namen für die Integer (z. B. 8 Bit „Byte“, 16 Bit „Word“ oder 32 Bit „Double Word“) werden üblicherweise Bezeichnungen wie int8 oder uint16 verwendet.
  + gebrochene Zahlen
    - Float
    - Double
  + verschiedene Varianten für unterschiedliche Wertebereiche und Genauigkeiten.

### Binär Minus

* Basis Prinzip
  + Ein Bit stell fest plus oder minus da
  + Leider minimiert das effektiven Speicher
* Potenz Prinzip
  + Zahl wird komplett negiert
    - alle 1 zu 0 und alle 0 zu 1
  + Letzter bit wird auf 1 gesetzt
  + Erster bit wird negativ gerechnet

### Binär Komma

* Die Eingabe - das Ergebnis einer Berechnung muss keine Dezimalzahl sein.
* Um reelle Zahlen direkt verarbeiten oder darstellen zu können, wurden Datentypen mit Komma Position benötigt.
* Festkommazahlen arbeiten mit einer vorab festgelegten Anzahl an Vor- und Nachkommastellen
* Festlegung der Stellenzahl begrenzt dieses Format und limitiert dadurch die Darstellung sehr großer oder sehr kleiner Zahlen.
* Eine universelle Anwendung ist mit Gleitkommazahlen durch die Darstellung in Exponentialschreibweise möglich.
* In dieser Form wird eine Zahl durch eine Mantisse m, eine Basis b und einem Exponenten e dargestellt.
  + Mantisse ist in der Regel eine Festkommazahl mit nur einer Ziffer (außer 0) vor dem Komma.
  + Für vorzeichenbehaftete Gleitkommazahlen wird noch ein Vorzeichenbit mitgeführt.
  + Das Vorzeichenbit / Exponentenbit kann die Werte 0 und 1 annehmen.
    - (-1)^1 = -1
    - (-1)^0 = 1
  + Es gibt 3 Formate
    - 32 bit
      * Der häufigste Datentyp ist Float
        + 1 Bit: Vorzeichen,
        + 8 Bit: Exponent und
        + 23 Bit: Mantisse.
    - 64 bit
      * Ein anderer gleitkomma Datentyp ist der Double
        + 1 Bit: Vorzeichen,
        + 11 Bit: Exponent und
        + 52 Bit: Mantisse.
    - 128 bit

### Fehler

* Fehler sind ganz allgemein Abweichungen von den Forderungen, also jedes unerwünschte abweichende Verhalten einer Software von den geforderten Parametern.
* Fehler passieren, man kann sie nicht ausschließen daher ist der umgang mit fehlern wichtiger als oder mindestens genauso wichtig wie das vorbeugen von Fehlern
* Fehler sind Teuer
* Zu Fehlern gehören nicht nur funktionale Fehler sondern beispielsweise auch Verhalten welche die effizienz der Laufzeit beinträchtigt und den Programmfluss verlangsamt.
* Grundliegend arbeitet man nach der Catch or Throw regel
* Catch or Throw Regel
  + (Catch) Entweder soll bei einen Fehler das Programm mit dem Fehler umgehen können indem es beispielsweise die Ausführung ändert
  + (Throw) Oder man soll einen neuen Fehler ausgeben, welcher den Anwender Hinweise auf die Stelle des Fehlers gibt.

### Redundanz

* Bei ausfall einer Sache gibt es immer mindestens eine zweite gleiche Instanz
* Anwendung in USV und Raid technologien
* ECC Memory chips