

## 2. Datentypen

---

### Beispiele elementarer Datentypen

- **Ganze Zahlen (integer)**

- Werte sind ganze Zahlen in vorgegebenen Bereich (z. B.  $-2^{31}$  bis  $2^{31} - 1$ )
- Übliche Operationen: Arithmetik (z. B. +, -, \*, Division mit Rest, Rest bei Division), Vergleiche (z. B. <, >, =, <=, >=), Zuweisungen, Bitoperationen

- **Unterbereiche**

- Teilbereich der ganzen Zahlen
- gleiche Operationen wie ganze Zahlen

- **Gleitkommazahlen**

- Werte sind rationale Zahlen mit abbrechender Dualdarstellung (Darstellung  $1.m * 2^e$  mit Binärzahlen Mantisse m und Exponent e)
- Übliche Operationen: Arithmetik, Vergleiche, Zuweisungen, oft auch sin, cos, exp, log,...

- **Festkommazahlen**

- Werte sind rationale Zahlen mit festem Dezimalpunkt und fester Nachkommastellenanzahl in festgelegtem Bereich (z. B. für Geldbeträge)
- Übliche Operationen: Arithmetik, Vergleiche, Zuweisungen

## 2. Datentypen

---

- **Komplexe Zahlen**

- Werte sind Paare von Fließkommazahlen, die Real- und Imaginärteil repräsentieren
- Übliche Operationen: Arithmetik, Zuweisung, Betrag, oft auch sin, cos, tan, exp,...

- **Rationale Zahlen**

- Werte sind Quotienten zweier ganzer Zahlen, dienen exakter Arithmetik ohne Rundungsfehler
- Übliche Operationen: Arithmetik, Vergleiche, Zuweisung

- **Aufzählungen**

- Werte stammen aus vom Programmierer frei festgelegter Menge
- Übliche Operationen: Zuweisung, Vergleiche (=, <>)

- **Boolescher Datentyp**

- Werte sind true und false. Datentyp dient oft als Ergebnis von Vergleichen.
- Übliche Operationen: Zuweisung, and, or, not, manchmal auch exklusives or (exor), not or (nor), not and (nand)

## 2. Datentypen

---

- **Zeichen**

- Werte sind einzelne Zeichen, oft aus ASCII-Tabelle, aber auch aus Unicode oder anderen Standardtabellen mit Sonderzeichen verschiedener Sprachen
- Übliche Operationen: Zuweisung, Vergleiche

## 2. Datentypen

---

### Beispiele für strukturierte Datentypen

- **Vektor oder Feld (array,...)**
  - festgelegte Komponentenanzahl (teilweise auch variabel)
  - homogen
  - Zugriff über Index
  - wahlfreier Zugriff auf Komponenten
  - sequentielle Organisation der Komponenten
  - Operationen: Zugriff über Index, Erstellen/Zerstören von Vektoren, zuweilen Operationen auf dem ganzen Vektor wie z. B. Zuweisung, Addition
- **multidimensionale Felder**
  - analog zu Feldern, aber Komponentenzugriff mit mehreren Indizes und Zugriff auf niederdimensionale Teilfelder

## 2. Datentypen

---

- **Datensätze (record, struct,...)**
  - festgelegte Komponentenanzahl
  - heterogen
  - Zugriff über Bezeichner
  - wahlfreier Zugriff auf Komponenten
  - sequentielle Organisation der Komponenten
  - Operationen: Zugriff über Bezeichner, Erstellen/Zerstören von Datensätzen, oft Zuweisung
- **Datensätze mit Varianten (z. B. in PASCAL)**
  - Typ der Komponenten verschiedener Datenobjekte darf verschieden sein
- **Listen**
  - variable Komponentenanzahl
  - oft heterogen, selten homogen
  - Zugriff ohne Bezeichner
  - sequentieller Zugriff auf Komponenten
  - Operationen: Sequentieller Zugriff, Einfügen/Löschen von Komponenten, Erstellen/Löschen der ganzen Liste

## 2. Datentypen

---

- **Stapel (stack)**
  - Liste mit Einfügen, Löschen und Zugriff auf Komponenten nur am Ende
- **Warteschlange (queue)**
  - Liste mit Einfügen nur am Ende, Zugriff und Löschen nur am Anfang
- **Zeichenketten (string)**
  - festgelegte oder variable Anzahl von Komponenten
  - homogen, nur Zeichen
  - Komponentenzugriff über Index
  - wahlfreier Zugriff
  - Operationen: Wahlfreier Zugriff, Erstellen/Löschen der ganzen Zeichenkette, Zuweisung, Verkettung zweier Zeichenketten, Vergleichsoperationen über Ordnung der Zeichen, Auswahl von Teilketten durch Positionierindizes, Ein-/Ausgabeformatierung, Auswahl von Teilketten durch Mustersuche

## 2. Datentypen

---

- **Zeiger und vom Programmierer gebildete Datenobjekte**
  - elementarer Datentyp Zeiger oder Referenz, der Adresse eines Datenobjekts oder Null enthält
  - Zeiger/Referenz kann nur auf Datenobjekt von festen Datentyp zeigen oder auf Datenobjekt beliebigen Datentyps (sprachabhängig)
  - Übliche Operationen: Erstellungsoperation zum Erzeugen von Datenobjekten fester Größe, Referenzierungsoperation, Zuweisung, oft Entfernungsoperation zum Löschen zuvor erzeugter Datenobjekte
- **Mengen (set)**
  - variable Komponentenanzahl
  - homogen
  - keine Bezeichner
  - nur Test auf Mitgliedschaft, oft auch Durchlauf aller Komponenten (Elemente)
  - ungeordnete Organisation der Komponenten
  - Operationen: Test auf Mitgliedschaft, Einfügen noch nicht vorhandener Werte, Löschen vorhandener Werte, Erstellen/Zerstören von Mengen, Schnitt, Vereinigung, mengentheoretische Differenz, oft Durchlauf aller Komponenten

## 2. Datentypen

---

- **Ausführbare Datenobjekte (z. B. in LISP)**
  - Wenn Programme in einer Sprache als Datenstruktur (z. B. Liste) behandelt werden können, können solche Datenobjekte ausgeführt werden.
  - Übliche Operationen: Operationen des zugrundeliegenden Datentyps, Ausführen
- **Dateien und Ein-/Ausgabe (= sequentielle Datei von Zeichen)**
  - variable Komponentenanzahl
  - homogen
  - bei sequentiellen Dateien nur sequentieller Zugriff, bei Direktzugriffsdateien beliebiger Zugriff über Index (Schlüsselwert), bei indexierten sequentiellen Dateien beliebiger Zugriff und Durchlaufen aller Komponenten
  - sequentielle Organisation der Komponenten
  - Übliche Operationen: Öffnen (entspricht oft auch dem Erstellen falls nicht vorhanden), Schließen, Löschen, Lesen einer Komponente, Schreiben einer Komponente, Dateiendetest