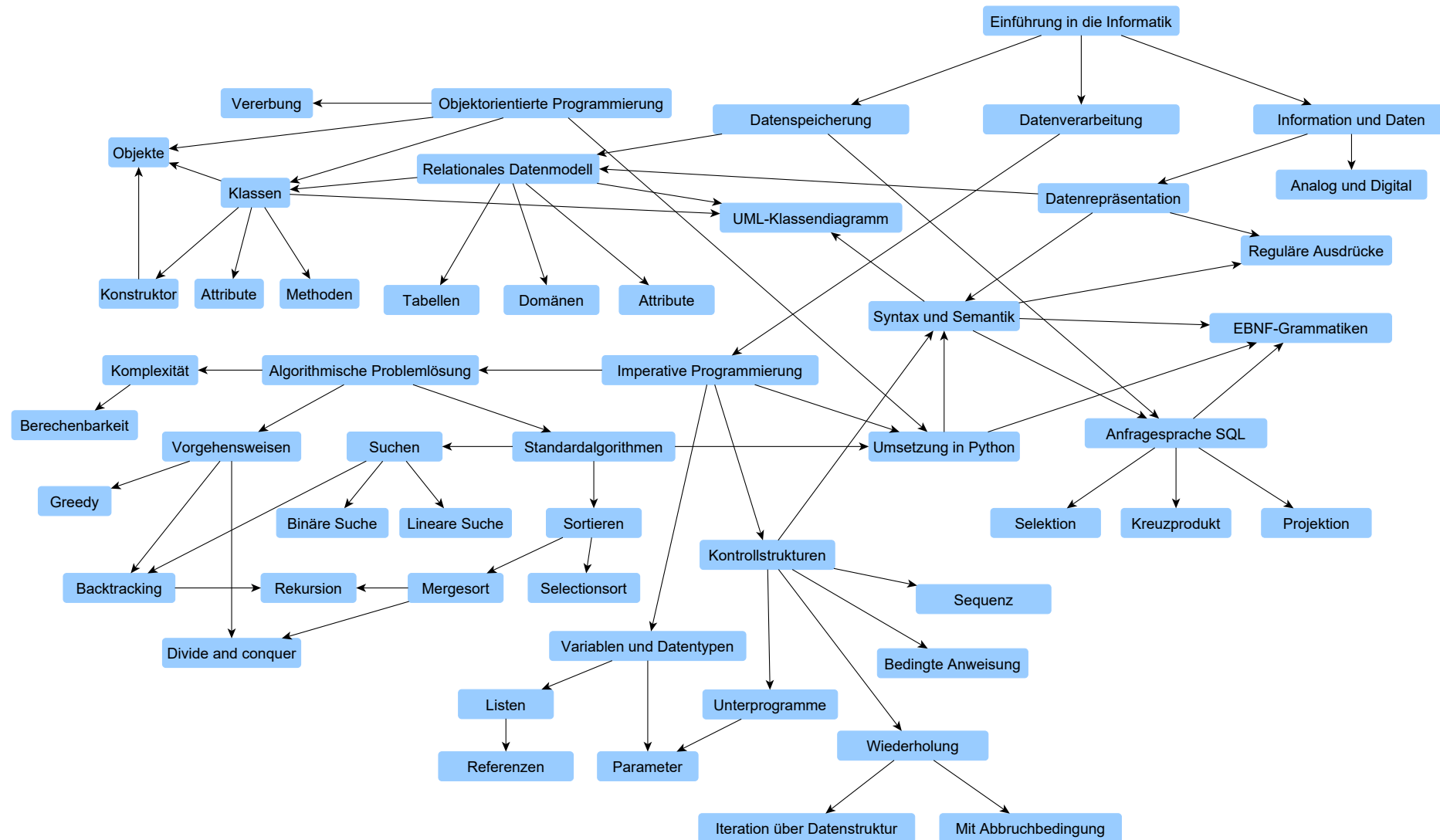


# Geschafft!



# Rückblick

Automatisierte Informationsverarbeitung bedeutet eigentlich automatisierte Datenverarbeitung, denn:

- Daten sind repräsentierte Information und
- die Deutung kommt (praktisch immer) von außerhalb.

Computersysteme verarbeiten ausschließlich Binärdaten.

Alle digital dargestellten Informationen lassen sich verlustfrei binär darstellen, analoge Informationen lassen sich (mit Informationsverlust) digitalisieren.

Die Struktur von Textdaten lässt sich mit regulären Ausdrücken oder Grammatiken beschreiben und überprüfen.

# Rückblick

Daten müssen zu ihrer Verarbeitung gespeichert werden.

Relationale Datenbanken sind auf diese Aufgabe hin optimiert, geben allerdings ein striktes Schema vor, wie Daten organisiert sein müssen (Relationen mit Attributen und Verweise über Schlüssel).

Mit der standardisierten Sprache SQL lassen sich Anfragen an Datenbanken formulieren, es können Daten extrahiert oder aggregiert werden. Das Ergebnis einer Anfrage ist immer eine Tabelle.

SQL erlaubt auch das Manipulieren der Daten (einfügen, verändern, löschen).

# Rückblick

Computersysteme arbeiten Befehlsorientiert.

Ein Prozessor führt immer nur den „Befehlszyklus“ aus (Befehl aus Speicher holen, ausführen, nächsten Befehl ermitteln). Die Befehle sind als Programme im Speicher abgelegt und damit selbst nur Daten.

Imperative Programmiersprachen bilden die Befehlsorientierung der Maschine mit einer besser Menschen-lesbaren Syntax ab.

Ein imperatives Programm wird als Sequenz von Anweisungen von oben nach unten ausgeführt. Bedingte Anweisung, Wiederholung und Unterprogramme sind die einzigen Möglichkeiten, davon abzuweichen.

# Rückblick

Jedes (imperative) Programm stellt einen Algorithmus dar.

Für algorithmisch lösbare Probleme gibt es viele mögliche Lösungen.

Die Laufzeitkomplexität ist ein Qualitätsmerkmal anhand derer sich Lösungen vergleichen lassen.

Sie beschreibt das asymptotische Wachstum der Laufzeit in Abhängigkeit der Eingabegröße und wird als (möglichst einfache) Funktion angegeben.

Es gibt typische Lösungsstrategien, die man immer wieder in der Programmierung verwendet: Exhaustive Suche und Greedy sind Beispiele dafür.

Backtracking implementiert eine exhaustive Suche rekursiv.

# Rückblick

Objektorientierte Programmierung erweitert das imperative Paradigma um die Möglichkeit, eigene Datentypen als Klassen zu definieren und diese mit Attributen und Operationen (Methoden, ggf. Operatoren) zu versehen.

Objektorientierung erlaubt Modularisierung und eine konzeptuell bessere Anbindung an relationale Datenbanken.

# Ausblick

Neben der imperativen und Objekt-orientierten Programmierung gibt es die deklarative Programmierung. Dort wird (ähnlich zu SQL) eher ein Ergebnis formuliert als ein spezifischer Weg dorthin.

Moderne Programmiersprachen weisen meistens Elemente aus allen drei Paradigmen auf, der Großteil ist allerdings Objekt-orientiert/imperativ.

Gutes Programmieren erfordert neben Syntax und Semantik auch Pragmatik.

Es gibt viele weitere Aspekte der Informatik (z.B. Netzwerke, maschinelles Lernen) die hier nicht behandelt wurden.