

|                              |                                   |                   |
|------------------------------|-----------------------------------|-------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> | <b>Anlage 7.1</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |                   |

# Formelsammlung

## 1.5.2 TG Informationstechnik

Version: V 4.30

Gültig ab Abitur 2024

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## Inhaltsverzeichnis:

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Beschreibung von Systemzuständen mit UML-Zustandsdiagrammen.....</b>   | <b>4</b>  |
| 1.1      | UML-Zustandsdiagramme (allgemein) .....   | 4         |
| 1.2      | Begriffserklärung für UML-Zustandsdiagramme .....   | 5         |
| 1.3      | Ergänzungen für Mikrocontroller.....  | 5         |
| 1.3.1    | Zustandsdefinition in C/CPP .....   | 5         |
| 1.3.2    | Zustandsvariable C/CPP .....  | 5         |
| 1.3.3    | Der Start-Pseudozustand .....   | 6         |
| 1.3.4    | Verhalten.....  | 6         |
| 1.3.5    | Zustandsübergang mit Wächterbedingung .....   | 6         |
| 1.3.6    | Zustandsübergang mit Ereignis und Wächterbedingung .....  | 7         |
| 1.3.7    | Selbsttransition und internes Ereignis .....  | 7         |
| 1.3.8    | Varianten von Transitionen .....  | 8         |
| <b>2</b> | <b>Hardware - Digitaltechnik.....</b>   | <b>9</b>  |
| 2.1      | Logikgatter .....   | 9         |
| 2.2      | Schaltnetze .....   | 10        |
| 2.3      | Schaltwerke .....   | 11        |
| 2.3.1    | Taktgenerator.....  | 11        |
| 2.3.2    | Flip-Flops .....  | 11        |
| 2.3.3    | RAM .....   | 11        |
| 2.3.4    | ROM .....   | 11        |
| 2.3.5    | Schieberegister .....   | 12        |
| 2.3.6    | Zähler (Blockschaltbild) .....  | 12        |
| 2.3.7    | Zähler (4-Bit) .....  | 12        |
| 2.4      | Sensoren.....   | 12        |
| 2.5      | Aktoren.....  | 13        |
| <b>3</b> | <b>Hardware - Mikrokontrollertechnik.....</b>   | <b>13</b> |
| 3.1      | Blockschaltbild „Prüfungscontroller“ .....  | 13        |
| 3.2      | Prozessorarchitektur .....  | 14        |
| 3.2.1    | Programmiermodell .....   | 14        |
| 3.2.2    | Prozessorkern CPU .....   | 14        |
| 3.2.3    | Blockschaltbild Mikrocontroller .....   | 15        |
| 3.2.4    | Befehlspipeline einer RISC-CPU .....  | 15        |
| 3.2.5    | Speicherarchitektur .....   | 15        |
| 3.3      | Onchip Peripherie .....   | 16        |
| 3.3.1    | Externer Interrupt .....  | 16        |
| 3.3.2    | Timer.....  | 16        |
| 3.3.3    | Puls-Weiten-Modulation (PWM) .....  | 17        |
| 3.3.4    | Analog – Digital – Wandlung .....   | 17        |
| 3.3.5    | Digital – Analog – Wandlung .....   | 17        |
| 3.4      | Externe Kommunikationsmöglichkeiten .....   | 18        |
| 3.4.1    | Serial Peripheral Interface (SPI) .....   | 18        |
| 3.4.2    | Universal Asynchronous Receiver Transmitter (UART) .....  | 18        |
| 3.4.3    | Inter-Integrated Circuit (I <sup>2</sup> C) SCL (Serial Clock): Taktleitung SDA (Serial Data): Datenleitung ..... | 19        |
| 3.5      | Glossar.....  | 20        |
| <b>4</b> | <b>Programmentwicklung und Objektorientierter Entwurf .....</b>   | <b>22</b> |
| 4.1      | Vergleichsoperatoren für Bedingungen (Pseudocode) .....   | 22        |
| 4.2      | Kontrollstrukturen (Pseudocode) .....   | 22        |
| 4.3      | Datentypen.....   | 23        |
| 4.3.1    | Elementare Datentypen.....  | 23        |
| 4.3.2    | Komplexe Datentypen .....   | 23        |

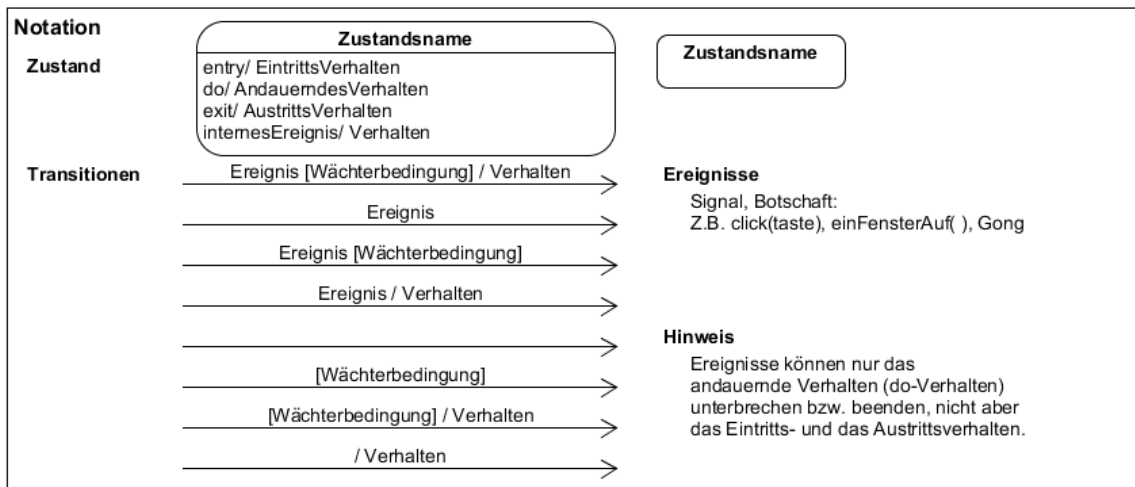
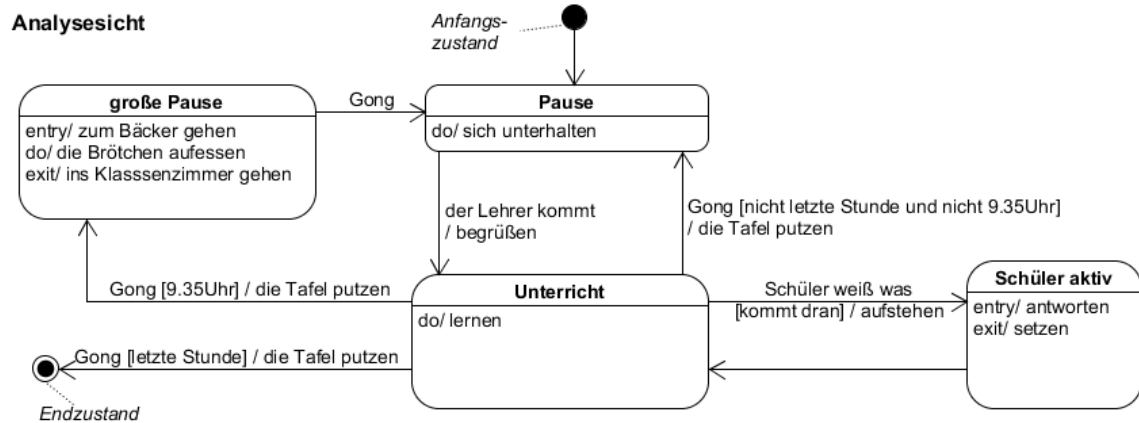
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

|          |  |           |
|----------|--|-----------|
| 4.4      | Klassen .....  | 24        |
| 4.4.1    | Attribute .....  | 24        |
| 4.4.2    | Operationen .....  | 25        |
| 4.4.3    | Assoziationen, Rollennamen und Multiplizitäten .....                   | 25        |
| 4.4.4    | Beispiel einer Operation mit einer Kollektion in Pseudocode .....      | 26        |
| 4.5      | Vererbung .....  | 26        |
| 4.6      | Abstrakte Klassen und Schnittstellen .....                             | 26        |
| 4.7      | Objektdiagramme .....  | 27        |
| 4.8      | Sequenzdiagramme .....   | 28        |
| 4.9      | Zustandsdiagramme .....  | 30        |
| <b>5</b> | <b>Datenstrukturen .....</b>   | <b>31</b> |
| 5.1      | Verkettete Liste .....   | 31        |
| 5.2      | Stapel .....   | 31        |
| 5.3      | Warteschlange .....  | 32        |
| 5.4      | Binärbaum .....  | 32        |
| 5.4.1    | Beispiel für einen Binärbaum der Tiefe 3 .....                         | 32        |
| 5.4.2    | Datenstruktur .....  | 33        |
| 5.4.3    | Operation ausgebenDatenInorder() der Klasse Knoten in Pseudocode ..... | 33        |
| <b>6</b> | <b>Künstliche Intelligenz .....</b>                                    | <b>34</b> |
| 6.1      | Klassifikation .....   | 34        |
| <b>7</b> | <b>Datenbanken .....</b>   | <b>35</b> |
| 7.1      | Datenbankmanagementsystem .....  | 35        |
| 7.2      | Entity-Relationship-Diagramm (ER-Diagramm) .....                       | 35        |
| 7.3      | Relationenmodell .....   | 36        |
| 7.4      | Abfrageformulierung mit SQL .....                                      | 36        |
| 7.4.1    | Projektion und Formatierung .....                                      | 36        |
| 7.4.2    | Selektion .....  | 37        |
| 7.4.3    | Verbund von Tabellen .....   | 38        |
| 7.4.4    | Aggregatfunktion .....   | 39        |
| 7.4.5    | Aggregatfunktion mit Gruppierung .....                                 | 40        |
| 7.4.6    | Selektion von Gruppen .....  | 40        |
| 7.4.7    | Komplette SQL-Anweisung .....  | 40        |
| <b>8</b> | <b>Vernetzte Systeme .....</b>   | <b>41</b> |
| 8.1      | Netzwerktechnik .....  | 41        |
| 8.1.1    | Netzwerksymbole .....  | 41        |
| 8.1.2    | Routing-Tabelle (IPv4) .....   | 41        |
| 8.1.3    | Aufbau IPv4-Adresse .....  | 42        |
| 8.1.4    | Aufbau IPv6-Adresse .....  | 42        |
| 8.2      | Schichtenmodelle .....   | 43        |
| 8.2.1    | ISO-OSI-7-Schichtenmodell .....  | 43        |
| 8.2.2    | TCP-IP-Schichtenmodell .....   | 43        |
| 8.3      | Header .....   | 43        |
| 8.3.1    | Ethernet II .....  | 43        |
| 8.3.2    | IPv4-Header .....  | 43        |
| 8.3.3    | IPv6-Header .....  | 44        |
| 8.3.4    | TCP –Header .....  | 44        |
| 8.3.5    | UDP –Header .....  | 44        |
| 8.4      | Internet der Dinge (IoT) .....   | 45        |
| 8.4.1    | MQTT-Protokoll (Message Queuing Telemetry Transport) .....             | 45        |
| 8.4.2    | HTTP-Protokoll (Hypertext Transfer Protocol) .....                     | 46        |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

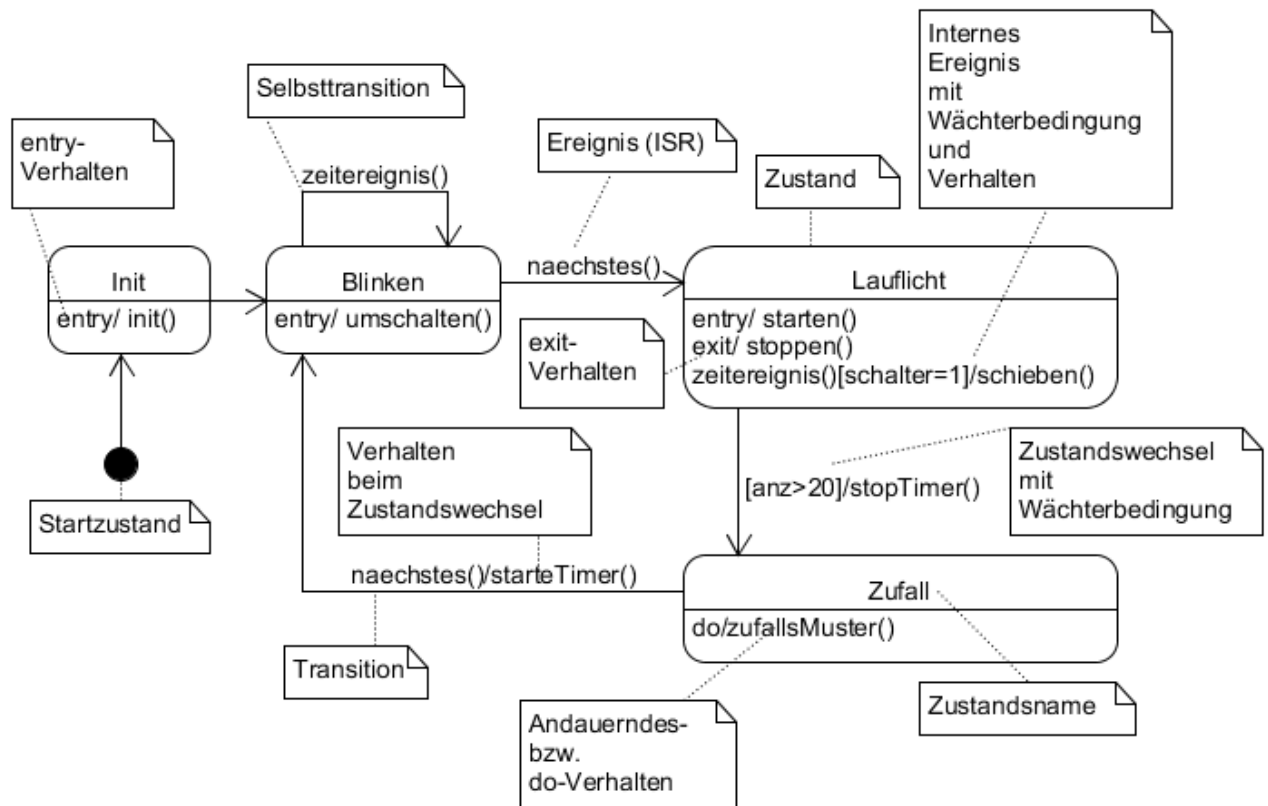
# 1 Beschreibung von Systemzuständen mit UML-Zustandsdiagrammen

## 1.1 UML-Zustandsdiagramme (allgemein)



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 1.2 Begriffserklärung für UML-Zustandsdiagramme



## 1.3 Ergänzungen für Mikrocontroller

Hinweis: Die folgenden Codebeispiele sind nicht verbindlich

### 1.3.1 Zustandsdefinition in C/CPP

Zustände sollten aus Gründen der Übersichtlichkeit Namen gegeben werden. Dadurch wird der Zusammenhang von Zustandsdiagramm und Programm verdeutlicht.

| Allgemein   | Beispiel                                      |
|---|---|
| #define Zustandsname Zustandsnummer                     | #define Init 0<br>#define Blinken 1           |
| oder  |   |
| enum zustandstyp<br>{Zustandsname=Zustandsnummer, ... } | enum zustandstyp<br>{Init=0, Blinken=1, ...}; |

### 1.3.2 Zustandsvariable C/CPP

Ein Zustand kann durch eine Zustandsvariable gekennzeichnet werden:

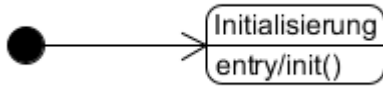
| Beispiele                    | Erklärung  |
|------------------------------|--|
| int zustand;                 | Zustandsvariable vom Typ int                     |
| PortOut zustand(PortC,0xFF); | Eine Portkonfiguration repräsentiert den Zustand |
| zustandstyp zustand;         | Zustandsvariable als enum (siehe oben)           |

Hinweis: Eine Zustandsvariable kann auch ein Ausgangsport des Mikrocontrollers sein (2. Beispiel). In diesem Fall bewirkt ein Zustandswechsel gleichzeitig, dass die Ausgänge entsprechend dem neuen Zustand angepasst werden.

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 1.3.3 Der Start-Pseudozustand

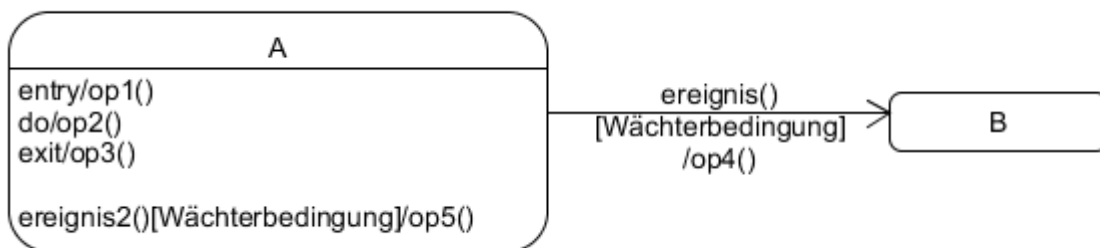
Die meisten Zustandsdiagramme beginnen mit einem Start-Pseudozustand:



Der ausgefüllte Kreis symbolisiert den Startpunkt des Zustandsdiagramms. Oft ist er mit dem Start des Mikrocontrollerprogramms gleich zu setzen. Die Transition vom Startpunkt zum ersten Zustand ist immer unbeschriftet.

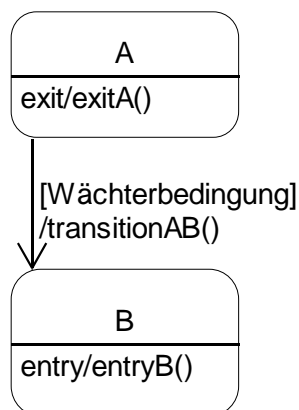
### 1.3.4 Verhalten

Verhalten sind Operationen oder Anweisungen, die an bestimmten Stellen des Zustandsdiagramms ausgeführt werden



| Verhalten                      | Ausführung  | Beispiel |
|--------------------------------|---|----------|
| Entry-Verhalten                | bei Eintritt in einen Zustand   | op1()    |
| Do-Verhalten                   | andauernd, solange der Zustand anhält   | op2()    |
| Exit-Verhalten                 | bei Verlassen des Zustands  | op3()    |
| Verhalten an der Transition    | beim Zustandswechsel  | op4()    |
| Verhalten am internen Ereignis | Wenn das interne Ereignis eintritt und gegebenenfalls eine Wächterbedingung erfüllt ist | op5()    |

### 1.3.5 Zustandsübergang mit Wächterbedingung



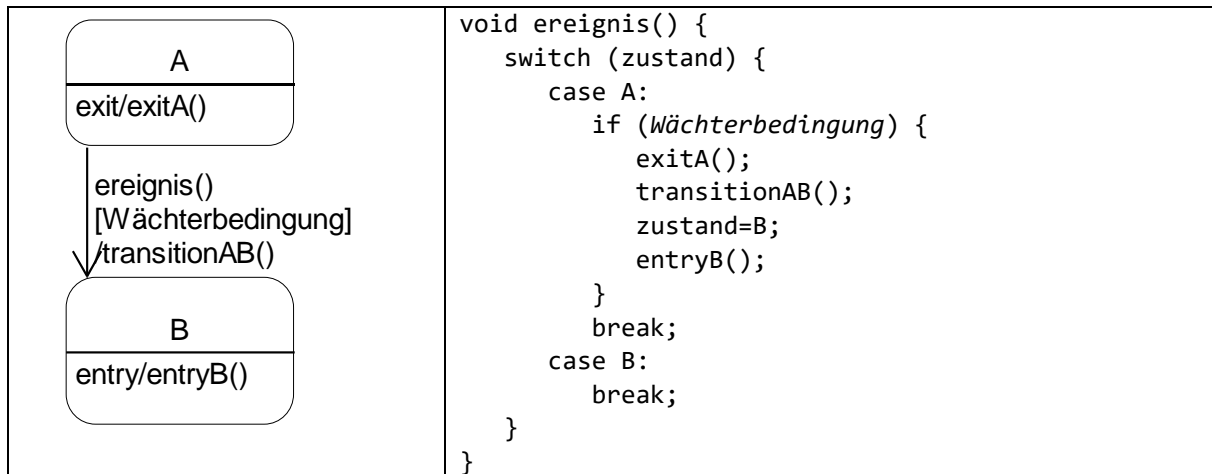
```
int main() {
    while(true) {
        switch (zustand) {
            case A:
                if (Wächterbedingung) {
                    exitA();
                    transitionAB();
                    zustand=B;
                    entryB();
                }
                break;
            case B:
                break;
        }
    }
}
```

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

In der Endlosschleife wird zuerst der Zustand geprüft. Falls sich der Mikrocontroller im Zustand A befindet, wird die Wächterbedingung an der Transition überprüft. Falls die Wächterbedingung erfüllt ist, erfolgt der Zustandswechsel. Es werden dann in folgender Reihenfolge die Verhalten ausgeführt:

1. Exit-Verhalten von Zustand A: `exitA()`
2. Verhalten an der Transition: `transitionAB()`
3. Zustandswechsel: `zustand=B`
4. Entry-Verhalten von Zustand B: `entryB()`

### 1.3.6 Zustandsübergang mit Ereignis und Wächterbedingung



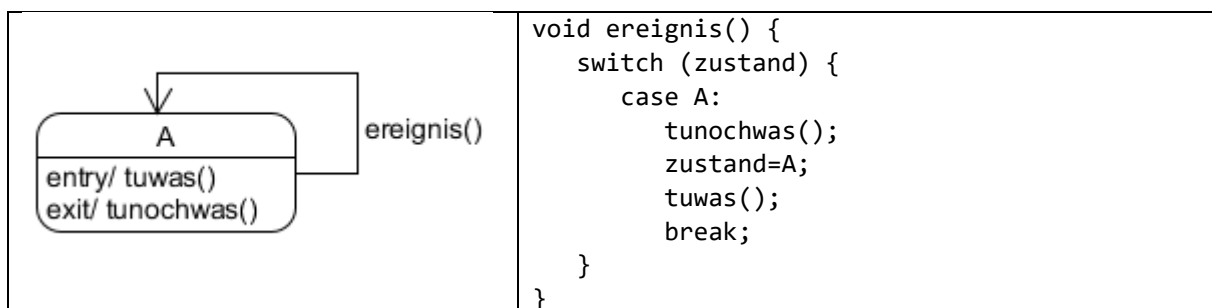
Es gibt **Aufruf-** und **Signal-Ereignisse**. Bei Signal-Ereignissen handelt es sich um Interrupts. Als Ereignisbezeichnung wird der Name der **Interrupt Service Routine (ISR)** verwendet.

In der ISR wird zuerst der Zustand geprüft. Falls sich der Mikrocontroller im Zustand A befindet, wird die Wächterbedingung an der Transition überprüft. Falls die Wächterbedingung erfüllt ist, erfolgt der Zustandswechsel. Es wird dann in folgender Reihenfolge das Verhalten ausgeführt:

1. Exit-Verhalten von Zustand A: `exitA()`
2. Verhalten an der Transition: `transitionAB()`
3. Zustandswechsel: `zustand=B`
4. Entry-Verhalten von Zustand B: `entryB()`

### 1.3.7 Selbsttransition und internes Ereignis

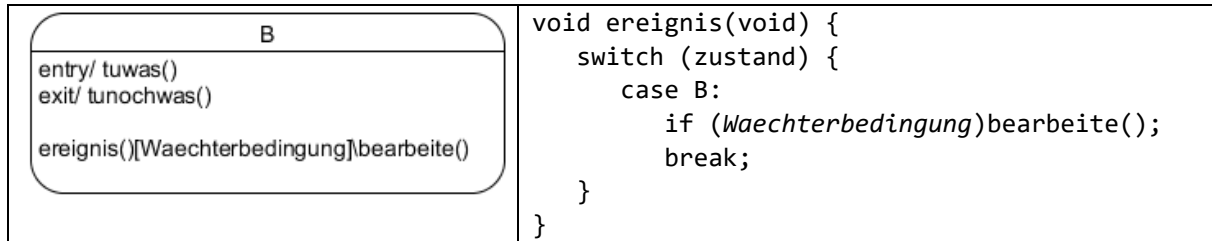
#### Selbsttransition



In der ISR `ereignis()` wird zuerst der Zustand geprüft. Falls sich der Mikrocontroller im Zustand A befindet, wird die Wächterbedingung, falls vorhanden, an der Transition überprüft. Falls die Wächterbedingung erfüllt ist, erfolgt der Zustandswechsel wieder nach A. Es werden nacheinander die `exit`-, `Transitions`- und `entry`-Verhalten ausgeführt.

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## Internes Ereignis



In der ISR ereignis() wird zuerst der Zustand geprüft. Falls sich der Mikrocontroller im Zustand B befindet, wird die Wächterbedingung, falls vorhanden, an der Transition überprüft. Falls die Wächterbedingung erfüllt ist, wird der Code, der zu diesem Ereignis in diesem Zustand gehört ausgeführt.

### 1.3.8 Varianten von Transitionen

Transitionen bezeichnen Zustandsübergänge und werden als Pfeil mit offener Spitze vom Ausgangszustand zum Zielzustand gezeichnet.








- |  |   |
|--|---|
|  | <p>a) <b>Transition ohne Beschriftung:</b> Sie bewirkt einen unmittelbaren Zustandswechsel, nachdem das entry-Verhalten beendet wurde.</p>  |
|  | <p>b) <b>Transition mit Wächterbedingung:</b> Sie bewirkt einen Zustandswechsel, sobald die Wächterbedingung erfüllt ist.</p>   |
|  | <p>c) <b>Transition mit Wächterbedingung und Verhalten:</b> Sie bewirkt einen Zustandswechsel, sobald die Wächterbedingung erfüllt ist. Beim Zustandswechsel wird das Verhalten ausgeführt.</p> |
|  | <p>d) <b>Transition mit Ereignis:</b> Sie bewirkt einen Zustandswechsel beim Eintreten des Ereignisses ereignis().</p>  |
|  | <p>e) <b>Transition mit Ereignis und Verhalten:</b> Verhalten wie bei d). Zusätzlich wird beim Zustandswechsel noch das Verhalten verhalten() ausgeführt.</p>                                   |
|  | <p>f) <b>Transition mit Ereignis und Wächterbedingung:</b> Verhalten wie bei d), falls zusätzlich die Wächterbedingung erfüllt ist.</p>   |
|  | <p>g) <b>Transition mit Ereignis, Wächterbedingung und Verhalten:</b> Verhalten wie bei f). Zusätzlich wird beim Zustandswechsel noch das Verhalten verhalten() ausgeführt.</p>                 |



|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 2 Hardware - Digitaltechnik

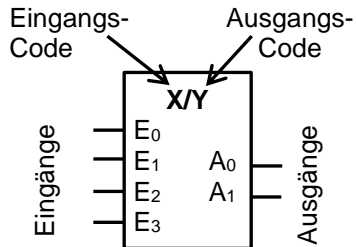
### 2.1 Logikgatter

| <div><div><div>NOT (Negation)</div><div><div><div>A</div><div></div><div>out</div></div></div><div><div><div>A</div><div><div>1</div></div><div><div>Y</div></div></div></div><div><div><math>Y = \bar{A}</math></div><div><math>Y = !A</math></div><div><math>Y = \neg A</math></div></div></div><div><table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></div></div>  | A | Y | 0 | 1 | 1 | 0 |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A   | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <div><div><div>AND (Konjunktion)</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>&amp;</div></div><div>Y</div></div></div><div><div><math>Y = A \wedge B</math></div><div><math>Y = A \&amp; B</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></div></div>   | B | A | Y | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | <div><div><div>NAND</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>&amp;</div></div><div><div>Y</div></div></div></div><div><div><math>Y = \overline{A \wedge B}</math></div><div><math>Y = !(A \&amp; B)</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></div></div>                                       | B | A | Y | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <div><div><div>OR (Disjunktion)</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>≥1</div></div><div>Y</div></div></div><div><div><math>Y = A \vee B</math></div><div><math>Y = A \# B</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></div></div>   | B | A | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | <div><div><div>NOR</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>≥1</div></div><div><div>Y</div></div></div></div><div><div><math>Y = \overline{A \vee B}</math></div><div><math>Y = !(A \# B)</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></div></div>   | B | A | Y | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <div><div><div>XOR (Antivalenz)</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>=1</div></div><div>Y</div></div></div><div><div><math>Y = (\bar{A} \wedge B) \vee (A \wedge \bar{B})</math></div><div><math>Y = (!A \&amp; B) \# (A \&amp; !B)</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></div></div> | B | A | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | <div><div><div>XNOR (Äquivalenz)</div><div><div><div>A</div><div>B</div><div></div><div>out</div></div></div><div><div><div>A</div><div>B</div><div><div>=</div></div><div>Y</div></div></div><div><div><math>Y = (\bar{A} \wedge \bar{B}) \vee (A \wedge B)</math></div><div><math>Y = (!A \&amp; !B) \# (A \&amp; B)</math></div></div></div><div><table><tr><th>B</th><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></div></div> | B | A | Y | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| B   | A | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

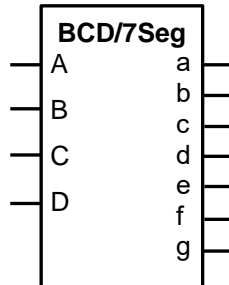
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 2.2 Schaltnetze

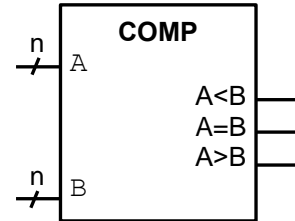
### Codeumsetzer (Umcodierer)



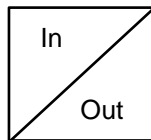
### BCD zu 7 Seg



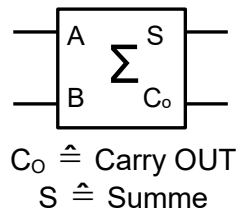
### Vergleicher



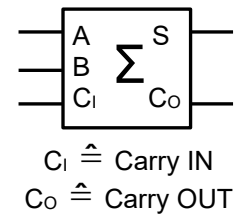
### BSB Codewandler



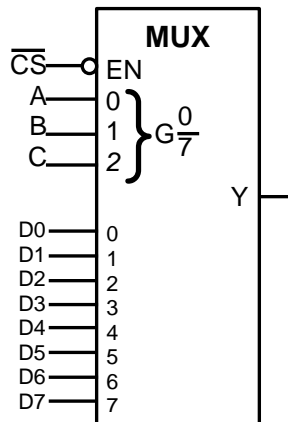
### Halbaddierer



### Volladdierer

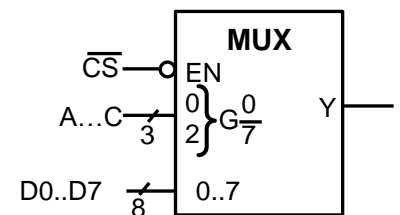


### MUX (8 zu 1)



| C | B | A | $\overline{CS}$ | Y  |
|---|---|---|-----------------|----|
| x | x | x | 1               | 0  |
| 0 | 0 | 0 | 0               | D0 |
| 0 | 0 | 1 | 0               | D1 |
| 0 | 1 | 0 | 0               | D2 |
| 0 | 1 | 1 | 0               | D3 |
| 1 | 0 | 0 | 0               | D4 |
| 1 | 0 | 1 | 0               | D5 |
| 1 | 1 | 0 | 0               | D6 |
| 1 | 1 | 1 | 0               | D7 |

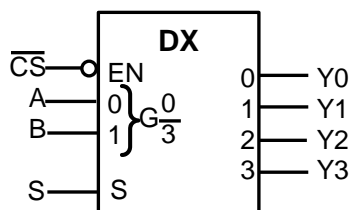
x  $\hat{=}$  don't care



Adress- und Datenleitungen können auch zusammen-gefasst werden

CS = chip select (low active)

### DEMUX (1 zu 4) Decodierer

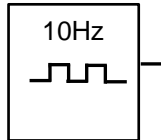


| B | A | $\overline{CS}$ | Y3 | Y2 | Y1 | Y0 |
|---|---|-----------------|----|----|----|----|
| X | X | 1               | 0  | 0  | 0  | 0  |
| 0 | 0 | 0               | 0  | 0  | 0  | S  |
| 0 | 1 | 0               | 0  | 0  | S  | 0  |
| 1 | 0 | 0               | 0  | S  | 0  | 0  |
| 1 | 1 | 0               | S  | 0  | 0  | 0  |

|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

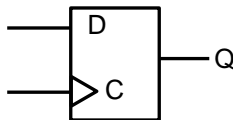
## 2.3 Schaltwerke

### 2.3.1 Taktgenerator



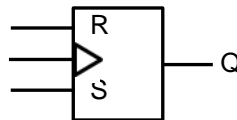
### 2.3.2 Flip-Flops

#### D-Flip-Flop



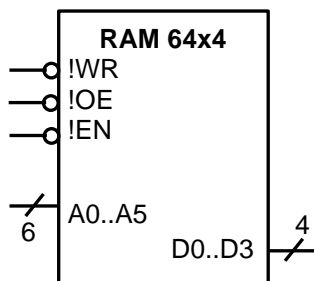
| Takt  | D | $Q^{n+1}$ |
|-------|---|-----------|
| ↑     | 0 | 0         |
| ↑     | 1 | 1         |
| sonst | X | $Q^n$     |

#### RS-Flip-Flop



| Takt  | R | S | $Q^{n+1}$   |
|-------|---|---|-------------|
| ↑     | 0 | 0 | $Q^n$       |
| ↑     | 1 | 0 | 0           |
| ↑     | 0 | 1 | 1           |
| ↑     | 1 | 1 | Undefiniert |
| sonst | x | x | $Q^n$       |

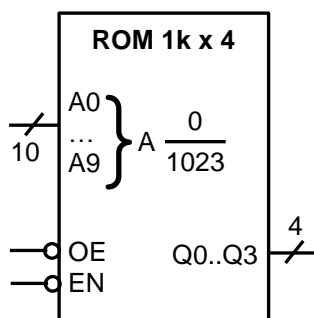
### 2.3.3 RAM



#### Schreib-Lese-Speicher mit 64 mal 4 Bit

- 4-Bit Registerbreite
- 64 Register gesamt
- **A0-A5**: Adresseingänge
- **D0-D3**: Ein-/Ausgabe des Speicherinhalts
- **WR=0**: lesen (von **D0-D3** in den Speicher)
- **WR=1**: schreiben (vom Speicher an **D0-D3**)
- **OE=1**: Tri-State
- **OE=0**: Speicherinhalt lesen
- **EN=0**: aktiviert den Baustein

### 2.3.4 ROM

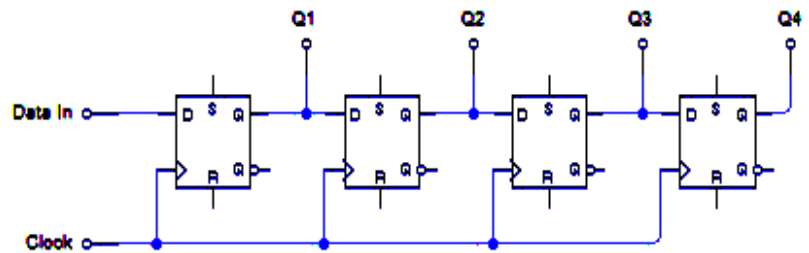
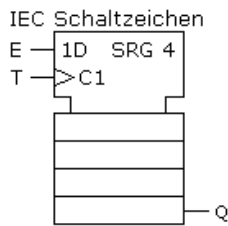


#### Festwertspeicher mit 1024 (1KiBi) mal 4 Bit

- **A0-A9**: Adresseingänge
- **OE=1**: Tri-State
- **OE=0**: Speicherinhalt lesen
- **EN=0**: aktiviert den Baustein
- **Q0-Q3**: Wert der Speicherzelle an Adresse A

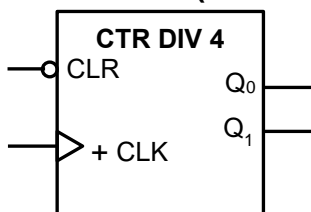
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 2.3.5 Schieberegister



Beispiel: Seriell In => Parallel Out

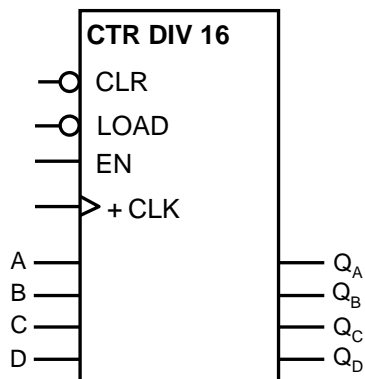
### 2.3.6 Zähler (Blockschaltbild)



Mit jeder steigenden Flanke an **CLK** wird der Zählerwert um 1 erhöht. Nach dem maximalen Wert wird der Zählwert wieder auf 0 gesetzt.

- **CTR**: Zähler (counter)
- **DIV 4**: 4 verschiedene binäre Zustände
- **CLR = 0** setzt den Counter auf den Wert **0** zurück
- **Q<sub>n</sub>** gibt den Zählerzustand aus

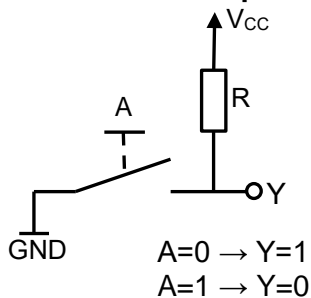
### 2.3.7 Zähler (4-Bit)



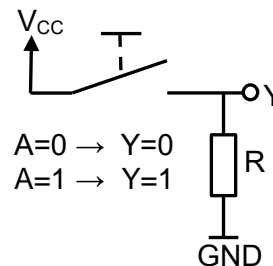
- **CTR**: Zähler (counter)
- **DIV 16**: 16 verschiedene binäre Zustände
- Vorwärtszähler (+)
- **EN = 1** und die positive Taktflanke führen zum nächsten Zählzustand
- Mit **LOAD = 0** kann ein Anfangszustand geladen werden
- **CLR = 0** setzt den Counter auf den Wert 0 zurück

## 2.4 Sensoren

#### Taster mit Pull-Up-Widerstand



#### Taster mit Pull-Down Widerstand

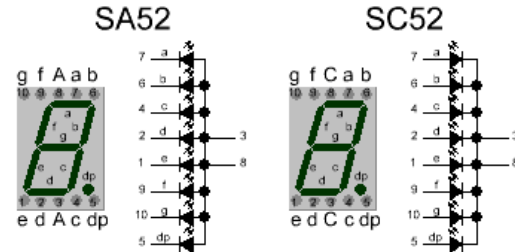


|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

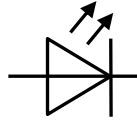
## 2.5 Aktoren

### 7-Segmentanzeige

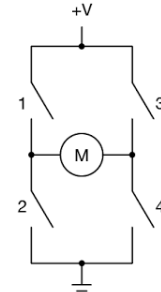
Common Anode ⇔ Common Cathode



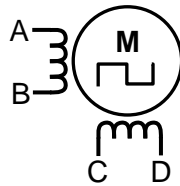
### LED



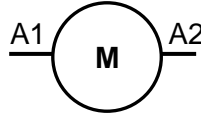
### H-Brücke



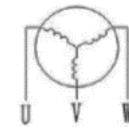
### Schrittmotor



### DC-Motor

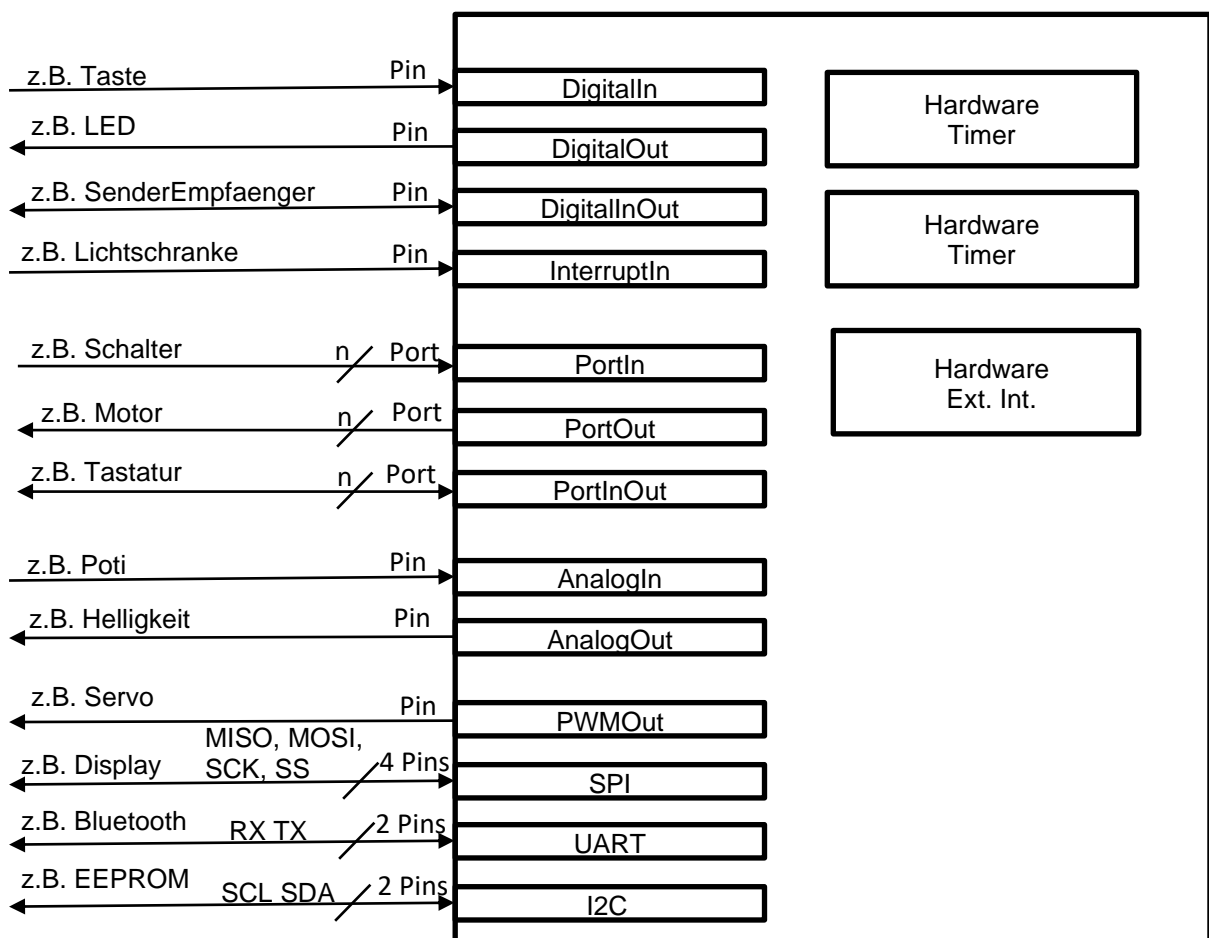


### BLDC-Motor



## 3 Hardware - Mikrocontrollertechnik

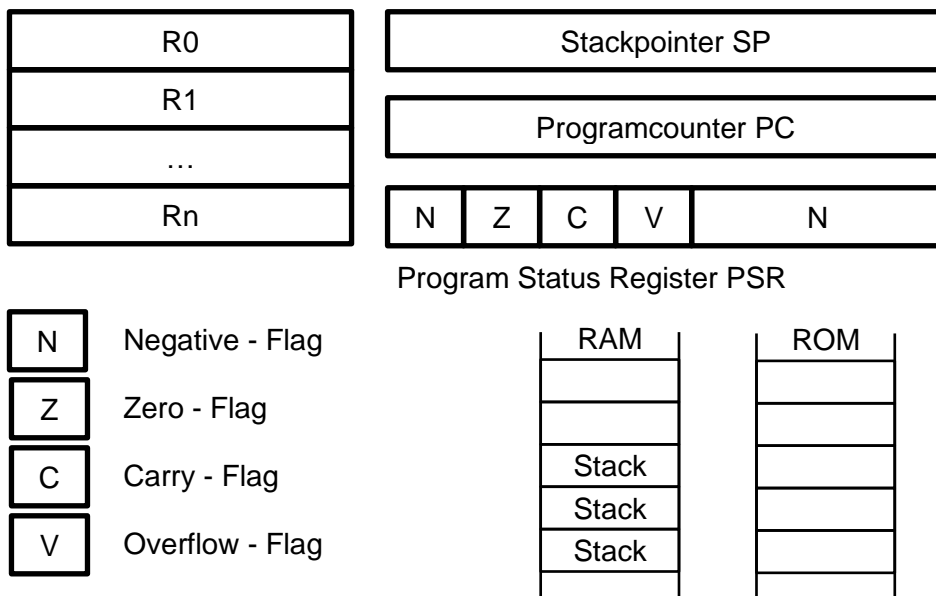
### 3.1 Blockschaltbild „Prüfungscontroller“



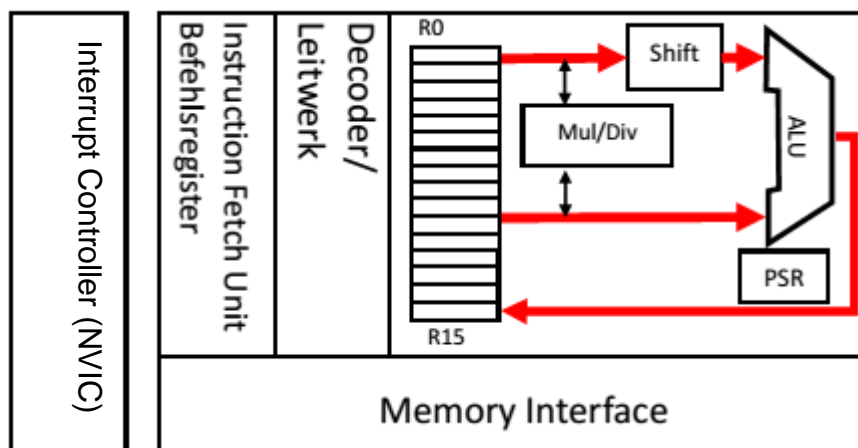
|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 3.2 Prozessorarchitektur

### 3.2.1 Programmiermodell

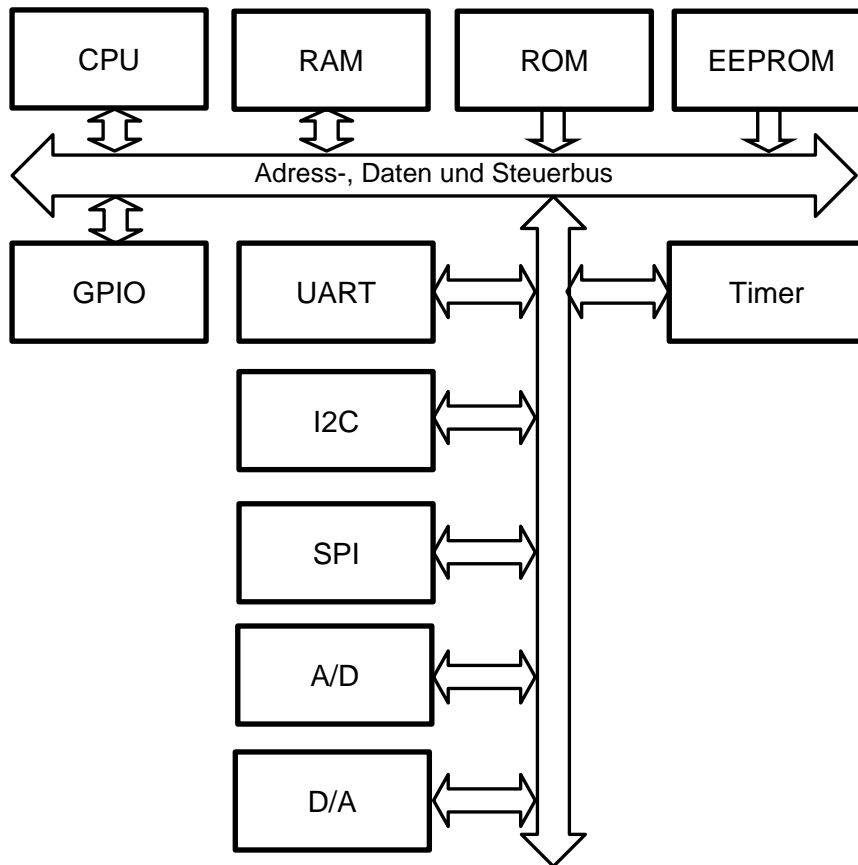


### 3.2.2 Prozessorkern CPU

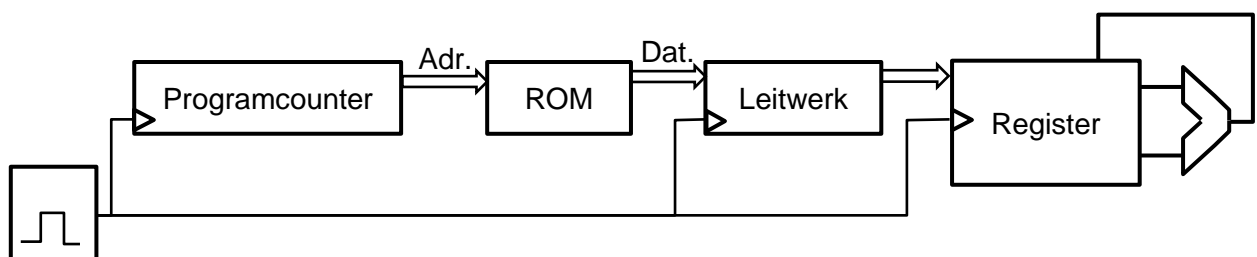


|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

### 3.2.3 Blockschaltbild Mikrocontroller

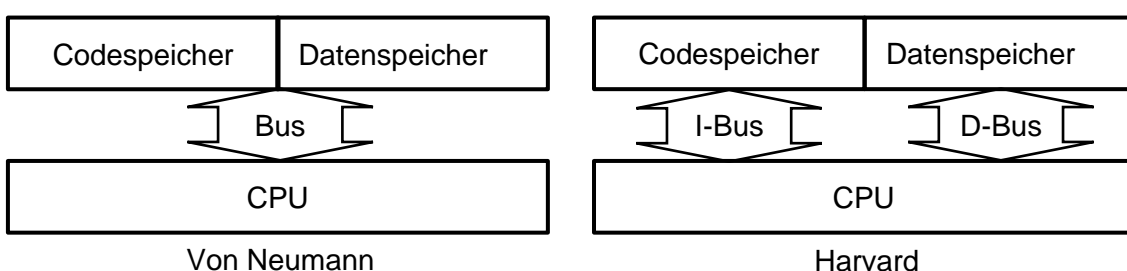


### 3.2.4 Befehlspipeline einer RISC-CPU



| Takt | 1       | 2        | 3         | 4         | 5         | 6         |
|------|---------|----------|-----------|-----------|-----------|-----------|
|      | Fetch 1 | Decode 1 | Execute 1 | Fetch 4   | Decode 4  | Execute 4 |
|      |         | Fetch 2  | Decode 2  | Execute 2 | Fetch 5   | Decode 5  |
|      |         |          | Fetch 3   | Decode 3  | Execute 3 | Fetch 6   |

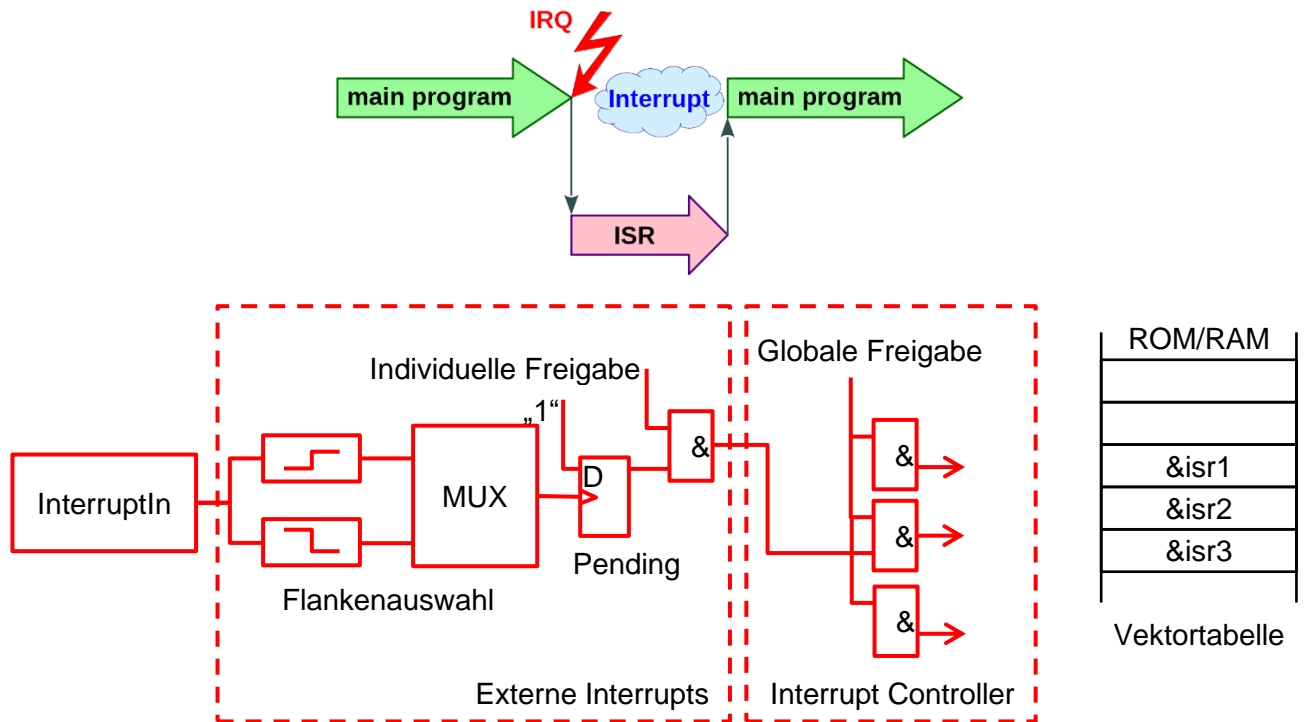
### 3.2.5 Speicherarchitektur



|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

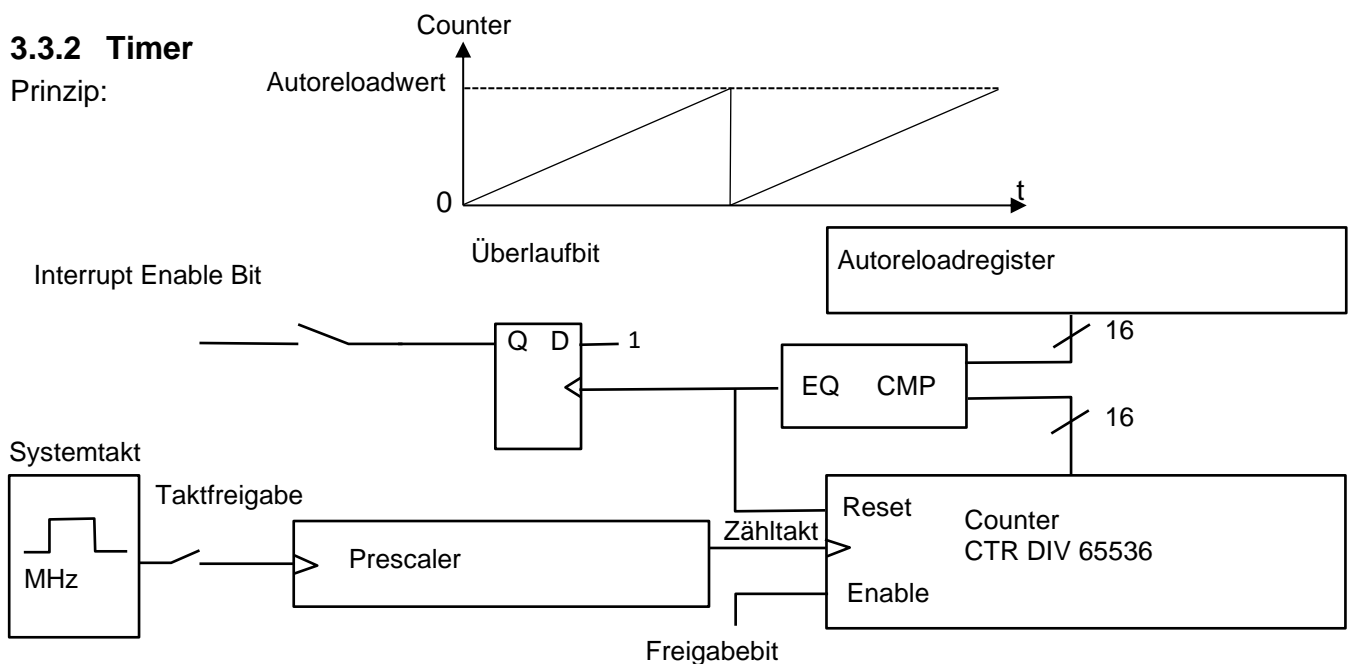
### 3.3 Onchip Peripherie

#### 3.3.1 Externer Interrupt



#### 3.3.2 Timer

Prinzip:

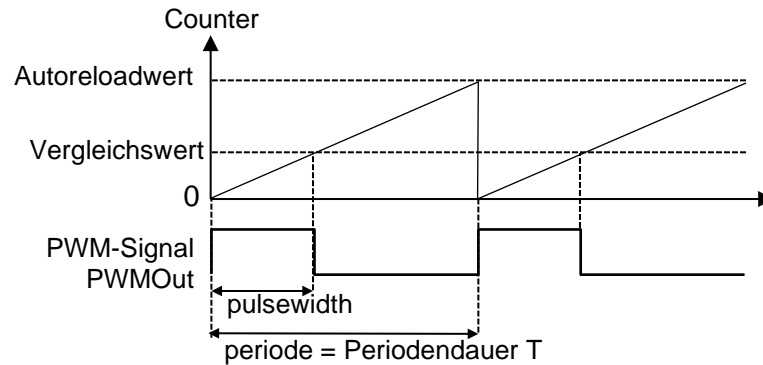




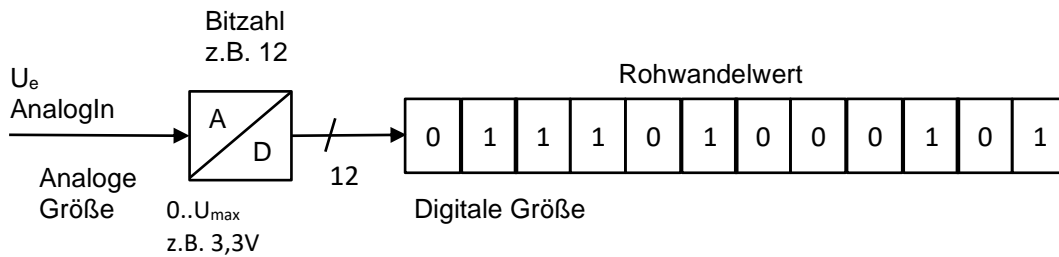
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 3.3.3 Puls-Weiten-Modulation (PWM)

Prinzip:



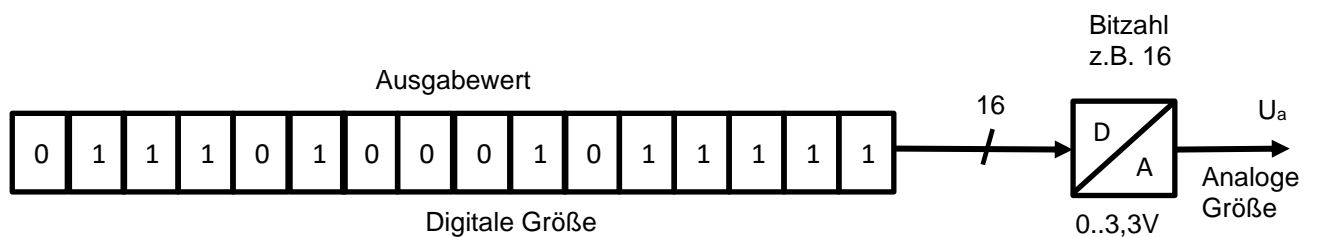
### 3.3.4 Analog – Digital – Wandlung



#### Berechnungsformeln

$$\begin{aligned} \text{Rohwandelwert} &= \frac{U_e}{U_{max}} \cdot (2^{\text{Bitzahl}} - 1) && \text{z.B. } \frac{U_e}{3,3V} \cdot 4095 \\ \text{Wandelwert als Kommazahl:} & x = (U_e/U_{max}) && \text{z.B. } x = (U_e/3.3) \\ \text{Stufung (analoge Auflösung):} & U_{max}/4095 && \text{z.B. } 3,3V/4095 \\ \text{Wandelwert als Ganzzahl linksbündig:} & \text{unsigned short } x = (U_e/U_{max}) * 65535 \end{aligned}$$

### 3.3.5 Digital – Analog – Wandlung



#### Berechnungsformeln

$$\begin{aligned} \text{float } x: & U_a = x \cdot 3,3V \\ \text{unsigned short } x: & U_a = \frac{x \cdot 3,3V}{65535} \end{aligned}$$

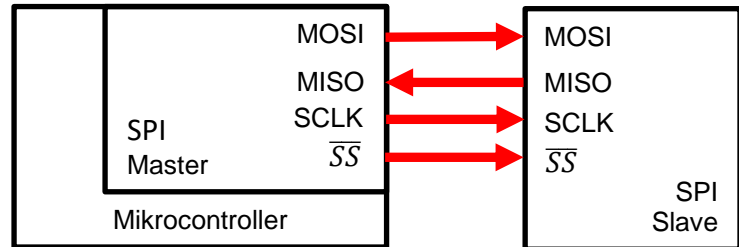
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 3.4 Externe Kommunikationsmöglichkeiten

#### 3.4.1 Serial Peripheral Interface (SPI)

Das **Serial Peripheral Interface (SPI)** dient der Kommunikation des Mikrocontrollers mit **Modulen** auf der Platine. Module sind

- Anzeigen,
- Speicher,
- LAN-Bausteine
- ...



#### Signale

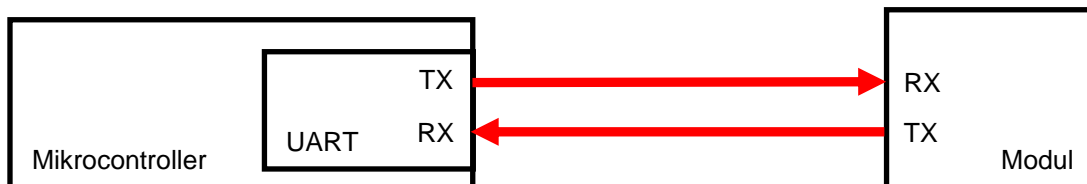
**MOSI** (Master Out Slave In): Sendeleitung

**MISO** (Master In Slave Out): Empfangsleitung

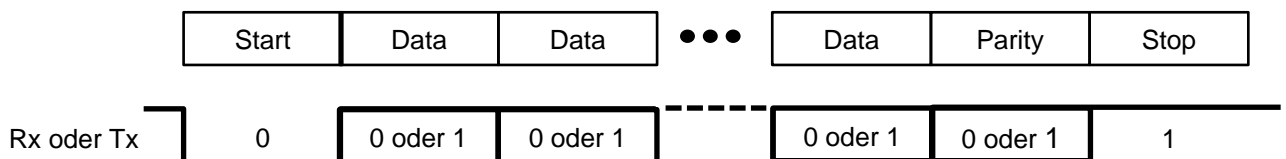
**SCLK** (Serial Clock): Taktleitung

**SS** (Slave Select): Auswahl des Slaves (Lowaktiv)

#### 3.4.2 Universal Asynchronous Receiver Transmitter (UART)



#### Frame



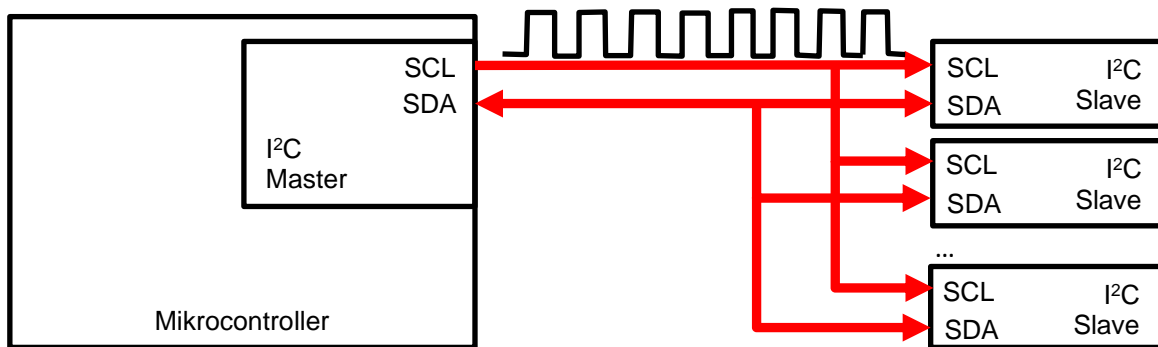
Eine UART-Übertragung beginnt immer mit einem Startbit (Low). Darauf folgen

- 5-8 **Datenbits** (Standard = 8)
- 0 oder 1 **Paritybit** (Standard = 0 none)
- 1 oder 2 **Stopbits** (Standard = 1)

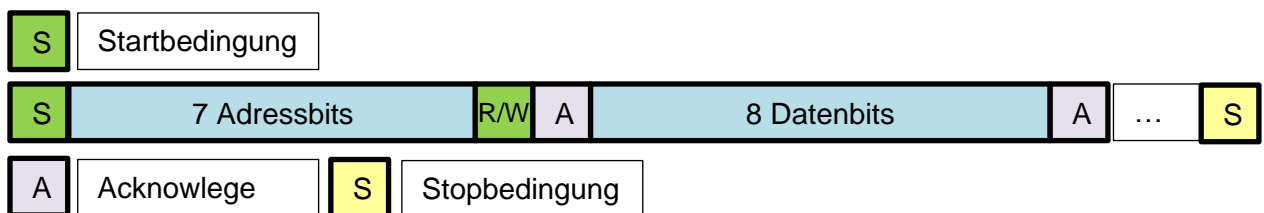
Falls ein Paritybit programmiert wurde, kann es gerade Parity (even) oder ungerade Parity (odd) anzeigen.

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 3.4.3 Inter-Integrated Circuit (I<sup>2</sup>C) SCL (Serial Clock): Taktleitung SDA (Serial Data): Datenleitung



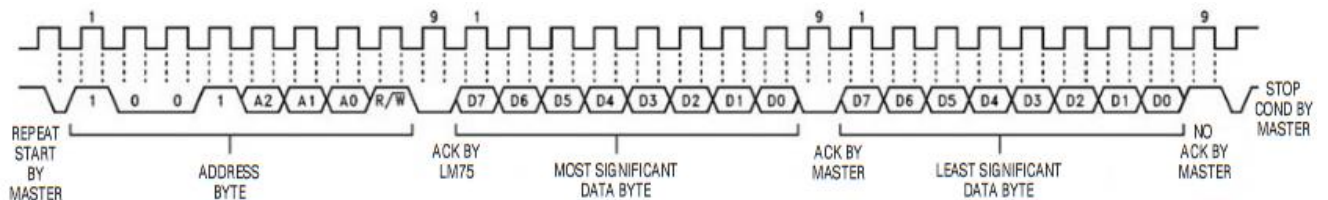
Frame:



Beispielhaft aufgeführte I<sup>2</sup>C-Bausteine bzw. Auszug Datenblätter Quelle: [www.alldatasheet.com](http://www.alldatasheet.com)

LM 75:

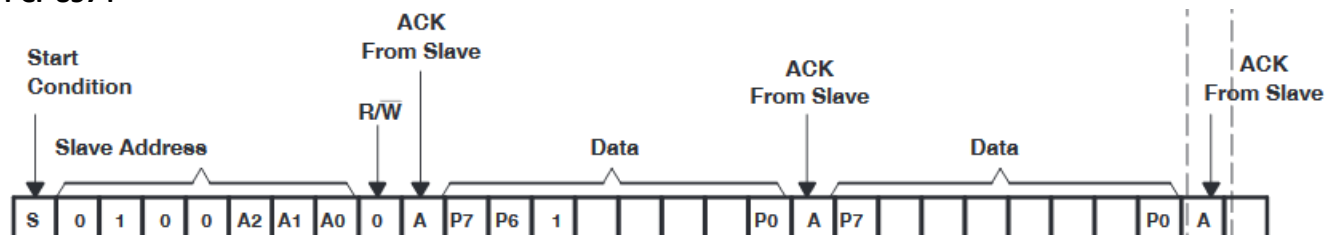
| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1     | 0     | 0     | 1     | A2    | A1    | A0    | RD/W  |



| UPPER BYTE                              |             |      |      |     |     |     |     | LOWER BYTE   |    |    |    |    |    |    |    |
|---|-------------|------|------|-----|-----|-----|-----|--------------|----|----|----|----|----|----|----|
| D15                                     | D14         | D13  | D12  | D11 | D10 | D9  | D8  | D7           | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Sign bit<br>1= Negative<br>0 = Positive | MSB<br>64°C | 32°C | 16°C | 8°C | 4°C | 2°C | 1°C | LSB<br>0.5°C | X  | X  | X  | X  | X  | X  | X  |

X = Don't care.

PCF 8574



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 3.5 Glossar

|                |   |
|----------------|---|
| Acknowledge    | Quittierung                                       |
| AD-Wandler     | Analog-Digital-Wandler                            |
| ALU            | Arithmetisch-Logische Einheit                     |
| AnalogIn       | Analogeingang Pin                                 |
| AnalogOut      | ,Analogausgang Pin                                |
| BCD            | Binär Codiert Dezimal                             |
| BLDC-Motor     | Bürstenloser Gleichstrommotor, Brushless DC-Motor |
| Bluetooth      | Funkstandard zur Datenübertragung                 |
| Carry          | Übertrag  |
| CISC           | Complex Instructionset Computer                   |
| CLK            | Clock, Takt                                       |
| CPU            | Central Processing Unit                           |
| CS             | Steuerleitung für Chip Select                     |
| CTR            | Counter   |
| DA-Wandler     | Digital-Analog-Wandler                            |
| DEMUX          | Demultiplexer                                     |
| DigitalIn      | Digitaleingang Pin                                |
| DigitalInOut   | Digital Input Output Pin bidirektional            |
| DigitalOut     | Digitalausgang Pin                                |
| DIV16          | Modulo 16   |
| EN             | Enable, Freigabe                                  |
| EPROM          | Erasable Programmable Read Only Memory            |
| EVA            | Eingabe Verarbeitung Ausgabe                      |
| Even           | gerade  |
| Frame          | Rahmen  |
| GPIO           | General Purpose Input Output                      |
| Hardware Timer | 16-Bit Timer                                      |
| I2C            | Inter-Integrated Circuit                          |
| InterruptIn    | Interrupteingang Pin                              |
| LED            | Light Emitting Diode Leuchtdiode                  |
| LOAD           | laden   |
| MISO           | MasterIn – SlaveOut                               |
| MOSI           | MasterOut – SlaveIn                               |
| MUX            | Multiplexer                                       |
| NVIC           | Nested Vector Interrupt Controller                |
| Odd            | Ungerade  |
| OE             | Steuerleitung für Output Enable                   |
| Overflow       | Überlauf  |
| Parity         | Geradzahligkeit                                   |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

|              |   |
|--------------|---|
| Periode      | Periodendauer   |
| PortIn       | Digitaleingang Port                                     |
| PortInOut    | Digital Input Output Port bidirektional                 |
| PortOut      | Digitalausgang Port                                     |
| Poti         | Potentiometer Einstellwiderstand für analoge Eingabe    |
| Pulsewidth   | Pulsbreite  |
| PWM          | Puls-Weiten-Modulation                                  |
| R0 usw.      | Prozessorregister                                       |
| RAM          | Random Access Memory                                    |
| RD           | Steuerleitung für lesen                                 |
| RISC         | Reduced Instructionset Computer                         |
| ROM          | Read Only Memory  |
| Rx           | Receive   |
| SCL(K)       | Serial Clock  |
| SDA          | Serial Data   |
| SPI          | Serial Peripheral Interface                             |
| SRG          | Schieberegister   |
| SS           | Slave Select  |
| Stack        | Stapel  |
| Stackpointer | Stapelzeiger  |
| Tx           | Transmit  |
| UART         | Universal Synchronous Asynchronous Receiver Transmitter |
| WR           | Steuerleitung für schreiben                             |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 4 Programmentwicklung und Objektorientierter Entwurf

### 4.1 Vergleichsoperatoren für Bedingungen (Pseudocode)

<, <=, >, >=, == oder =, ≠ oder !=

Anmerkung: Die Operatoren für Vergleiche und Wertzuweisungen müssen unterschieden werden können.

### 4.2 Kontrollstrukturen (Pseudocode)

#### Zuweisung

```
dieVariable ← derAusdruck
dieVariable := derAusdruck
dieVariable = derAusdruck
```

#### Sequenz

```
anweisung1
anweisung2
anweisung3
```

#### Auswahl

##### Einseitige Auswahl

```
WENN bedingung
    anweisung1
...
ENDE WENN
```

##### Zweiseitige Auswahl

```
WENN bedingung
    anweisungA1
...
SONST
    anweisungB1
...
ENDE WENN
```

##### Mehrfachauswahl

```
FALLS variable GLEICH
    bedingung1: anweisungA1
    ...
    bedingung2: anweisungB1
    ...
    bedingung3: anweisungC1
    ...
    SONST: anweisungD1
    ...
ENDE FALLS
```

#### Schleife (Iteration)

##### Schleife mit Eintrittsbedingung

```
SOLANGE bedingung
    anweisung1
...
ENDE SOLANGE
```

##### Schleife mit Austrittsbedingung

```
WIEDERHOLE
    anweisung1
...
SOLANGE bedingung
```

##### Zählschleife

```
FÜR i ← 0 BIS n SCHRITT s
    anweisung1
...
ENDE FÜR
```

##### Schleife über Kollektion

```
FÜR element IN kollektion
    anweisung1
...
ENDE FÜR
```

##### Schleife mit Abbruchbedingung

```
FÜR element IN kollektion
    anweisungA1
    ...
    WENN bedingung
        ABBRUCH
    ENDE WENN
    anweisungB1
    ...
ENDE FÜR
```

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 4.3 Datentypen

### 4.3.1 Elementare Datentypen

| Datentyp              | Abkürzungen                 | Werte                     |
|-----------------------|-----------------------------|---------------------------|
| Boolscher Datentyp    | Boolean, boolean, bool, ... | wahr, falsch, true, false |
| Ganzzahliger Datentyp | GZ, Integer, int, ...       | -24, 0, 123, ...          |
| Fließkomma-Datentyp   | FKZ, Real, double, ...      | -3.567, 0.0, 3.141, ...   |
| Zeichen-Datentyp      | Zeichen, char, ...          | 'Z', 'a', '&', ...        |
| Text-Datentyp         | Text, String, string, ...   | "Hello world!!!", ...     |

Für den Datentyp Text ist als Vergleichsoperator nur == bzw. = definiert. Außerdem kann der Operator + für die Verbindung von zwei Texten verwendet werden. Auch bei Texten muss der Vergleich und die Zuweisung eindeutig unterschieden werden können (vgl. 4.1).

### 4.3.2 Komplexe Datentypen

| Zeit  |
|---|
| ...   |
| +Zeit( )<br>+Zeit(pStunde:GZ,pMinute:GZ,pSekunde:GZ)<br>+gibStunde( ):GZ<br>+gibMinute( ):GZ<br>+gibSekunde( ):GZ<br>+istVor(pZeit:Zeit):Boolean<br>+istNach(pZeit:Zeit):Boolean<br>+zeitMinusSekunden(pSekunden:GZ):Zeit<br>+zeitPlusSekunden(pSekunden:GZ):Zeit<br>+gibText( ):Text |

| Datum  |
|--|
| ...  |
| +Datum( )<br>+Datum(pTag:GZ,pMonat:GZ,pJahr)<br>+gibTag( ):GZ<br>+gibMonat( ):GZ<br>+gibJahr( ):GZ<br>+istVor(pDatum:Datum):Boolean<br>+istNach(pDatum:Datum):Boolean<br>+anzahlTageBis(pDatum:Datum):GZ<br>+anzahlTageSeit(pDatum:Datum):GZ<br>+gibText( ):Text |

| Liste<Typ>  |
|---|
| ...   |
| +Liste<Typ>( )<br>+gibLaenge( ):GZ<br>+gib(pIndex:GZ):Typ<br>+ersetzen(pIndex:GZ,pElement:Typ)<br>+einfuegen(pIndex:GZ,pElement:Typ)<br>+anhaengen(pElement:Typ)<br>+verketten(pListe>Liste<Typ>)<br>+entfernen(pIndex:GZ):Typ<br>+entfernen(pElement:Typ)<br>+enthaelt(pElement:Typ):Boolean<br>+kopieren( ):Liste<Typ><br>... |

Listen beinhalten Daten vom gleichen Typ. Dabei kann es sich um elementare oder komplexe Datentypen (Klassen) handeln, z.B. Liste<GZ> oder Liste<Person>.

Die Operationen ersetzen und einfuegen unterscheiden sich dadurch, dass beim Ersetzen das Element am Index pIndex ersetzt wird und die Liste somit ihre Länge behält, während beim Einfügen die Liste verlängert wird, da das Element pElement die nachfolgenden Elemente um eine Position nach hinten verschiebt.

Die Operation entfernen ist überladen. Wird sie mit einer ganzzahligen Löschesposition als Argument aufgerufen, gibt die Operation das gelöschte Objekt vom Datentyp Typ zurück. Wird entfernen mit einem Argument vom Datentyp Typ aufgerufen, wird dieses Objekt in der Liste von vorne gesucht und das erste gefundene Objekt, falls vorhanden, gelöscht.

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### Alternative Notationen für Listen

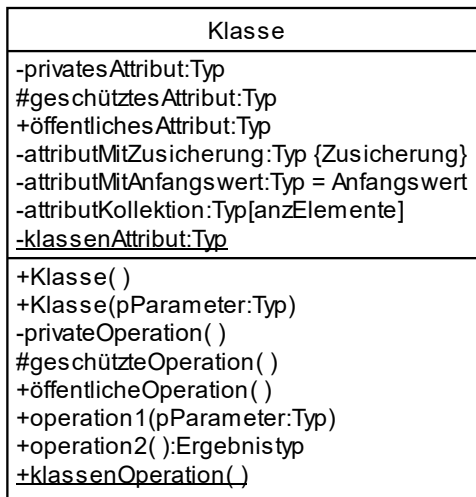
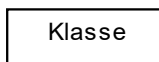
Liste highscore vom Datentyp Liste<GZ>

| Standardnotation   | Alternative Notation  | Bedeutung                         |
|--|-----------------------|-----------------------------------|
| highscore ← NEU Liste<GZ>()  | highscore ← []        | Leere Liste anlegen.              |
| highscore ← NEU Liste<GZ>()<br>FÜR i←0 BIS 2 SCHRITT 1<br>highscore.anhaengen(0) | highscore ← [0, 0, 0] | Liste mit drei Elementen anlegen. |
| h ← highscore.gib(0)   | h ← highscore[0]      | Element einer Liste lesen.        |
| highscore.ersetzen(3,5)  | highscore[3] ← 5      | Element einer Liste schreiben.    |

### Notationen für Felder

| Standardnotation       | Bedeutung                       |
|------------------------|---------------------------------|
| highscore ← NEU GZ[10] | Feld für 10 Highscores anlegen. |
| highscore[0] ← 15      | Ersten Highscore auf 15 setzen. |

## 4.4 Klassen



### 4.4.1 Attribute

Die Bezeichner von Attributen beginnen mit einem Kleinbuchstaben (vgl. UML-Standard). Attribute haben im Klassendiagramm folgenden Aufbau:

Sichtbarkeit bezeichner:Typ<[Multiplizität]>[=Anfangswert]<{Zusicherung}>

Die in spitzen Klammern notierten Inhalte, z.B. <[Multiplizität]>, sind optionale Bestandteile der Attribute.



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

| Sichtbarkeit | Zeichen |
|--------------|---------|
| privat       | -       |
| geschützt    | #       |
| öffentlich   | +       |

| Typ                         |
|-----------------------------|
| Elementarer Datentyp        |
| Komplexer Datentyp (Klasse) |

| Anfangswert  |
|--|
| Wert den das Attribut bei der Erzeugung des Objekts annimmt. |

| Zusicherung                                       |
|---|
| Vorschriften für Attribute {wert>0}, {read only}. |

#### 4.4.2 Operationen

Prozeduren bzw. Funktionen von Programmiersprachen nennt man im Kontext der Objektorientierung Operationen. Ihre Bezeichner starten, wenn möglich, mit einem Verb. Wie bei Attributen ist der erste Buchstabe ein Kleinbuchstabe. Operationen haben im Klassendiagramm folgenden Aufbau:

Sichtbarkeit operationsbezeichner(<Parameterliste>):<Rückgabetyp>

Eine Parameterliste kann leer sein oder einen oder mehrere Parameter enthalten. Die Parameter werden nach folgendem Schema definiert:

pName:Typ, ...

Die in spitzen Klammern notierten Inhalte, z.B. <Parameterliste>, sind optionale Bestandteile der Operationsdeklaration.

#### 4.4.3 Assoziationen, Rollennamen und Multiplizitäten



Gerichtete Assoziation

Bidirektionale Assoziation

| Multiplizität | Bedeutung      |
|---------------|----------------|
| 1             | genau 1        |
| 0..1          | 0 oder 1       |
| 3..6          | 3, 4, 5 oder 6 |
| *             | 0 bis viele    |
| 2..*          | 2 bis viele    |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

#### 4.4.4 Beispiel einer Operation mit einer Kollektion in Pseudocode

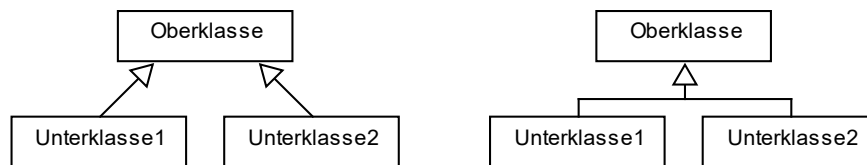
OPERATION anlegenPerson(pName:Text,personen:Liste<Person>):Boolean

Lokale Variablen: gefunden:Boolean, neuePerson:Person, person:Person

```

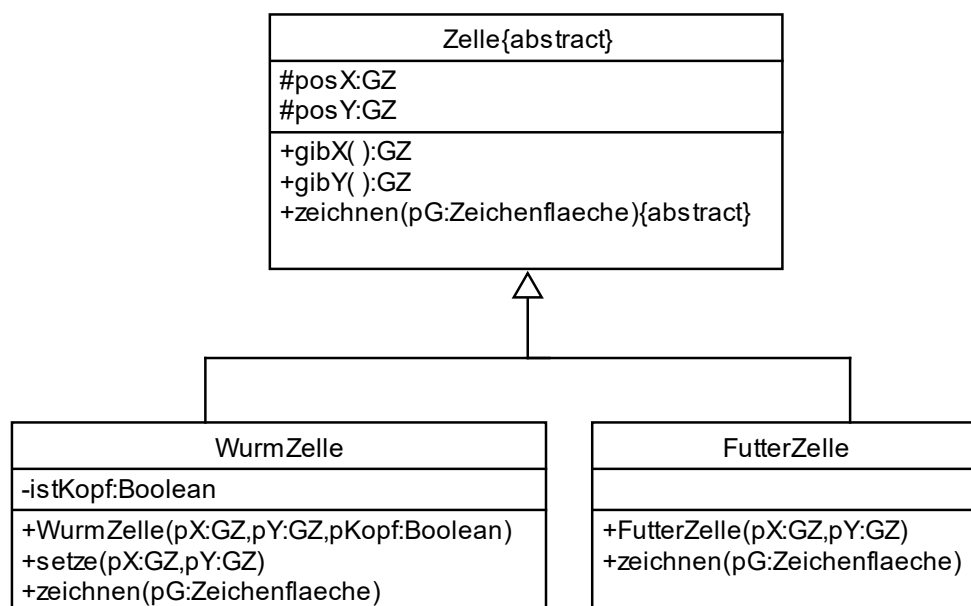
gefunden ← falsch
FÜR person IN personen
    WENN person.gibName() = pName
        gefunden ← wahr
        ABBRUCH
    ENDE WENN
ENDE FÜR
WENN gefunden = falsch
    neuePerson ← NEU Person(pName)
    personen.anhaengen(neuePerson)
ENDE WENN
RÜCKGABE gefunden
    
```

#### 4.5 Vererbung

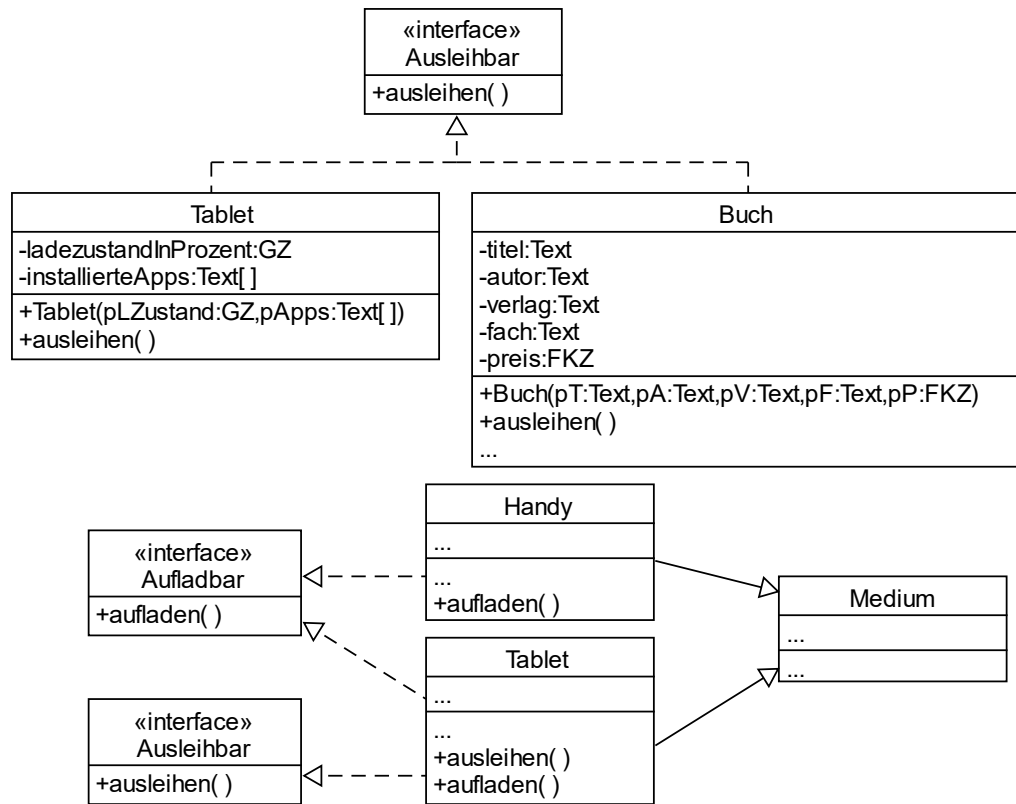


Oberklassen sind Generalisierungen und Unterklassen Spezialisierungen.

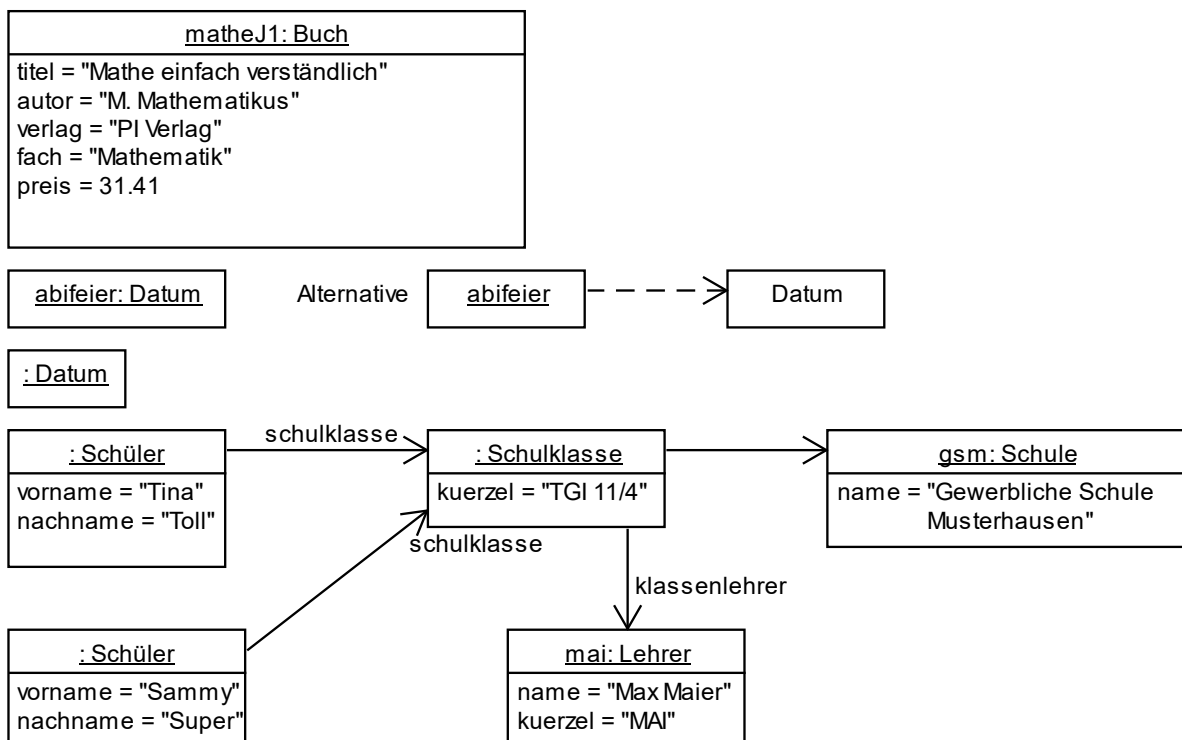
#### 4.6 Abstrakte Klassen und Schnittstellen



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |



#### 4.7 Objektdiagramme



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

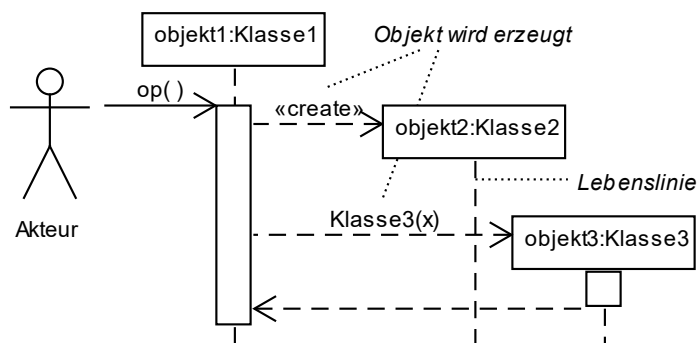
## 4.8 Sequenzdiagramme

Allgemeines:

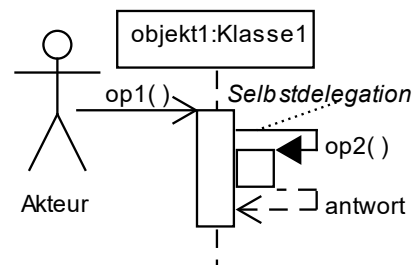
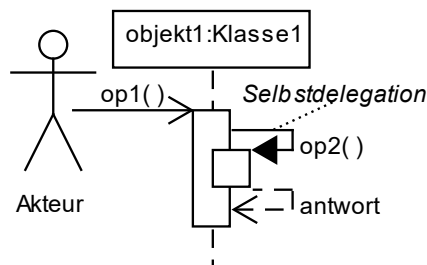
- Es wird nicht zwischen unterstrichenen und nicht-unterstrichenen Objekten im Sequenzdiagramm unterschieden.

### Erzeugung von Objekten

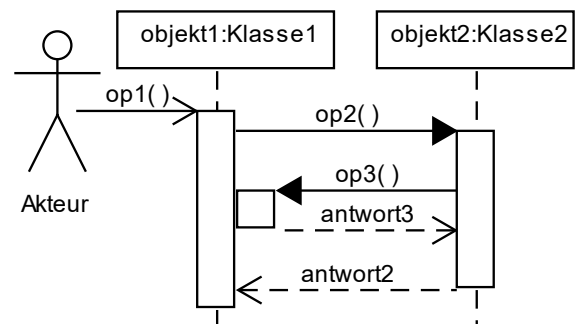
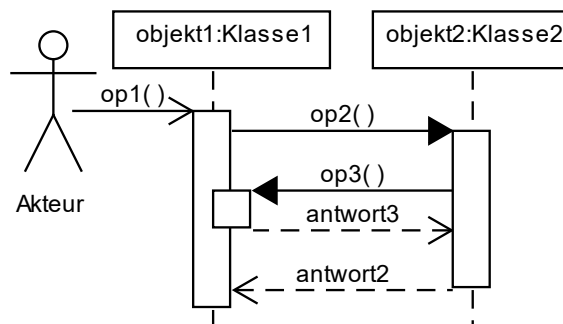
Ein Objekt kann im Sequenzdiagramm immer mit einem spezifischen Konstruktor erzeugt werden. Ist die Auswahl des Konstruktors nicht bedeutsam, so wird die Objekterzeugung durch `<<create>>` dargestellt.



### Selbstdelegation (alternative Darstellungen)

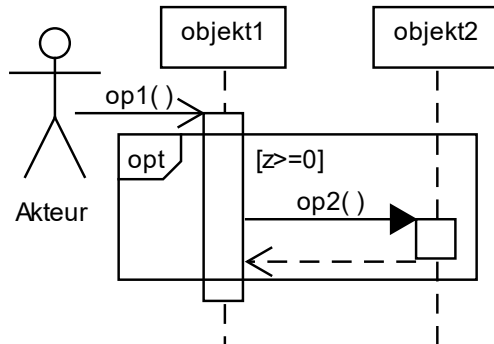


### Wechselseitige Botschaften (alternative Darstellungen)

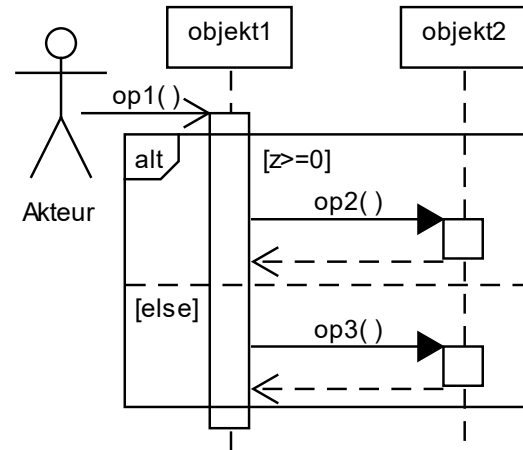


|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

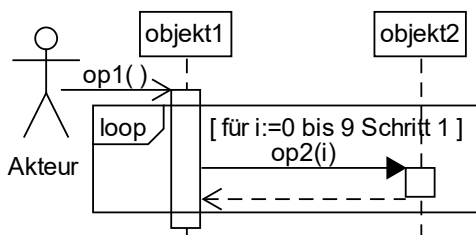
### Option – einseitige Verzweigung



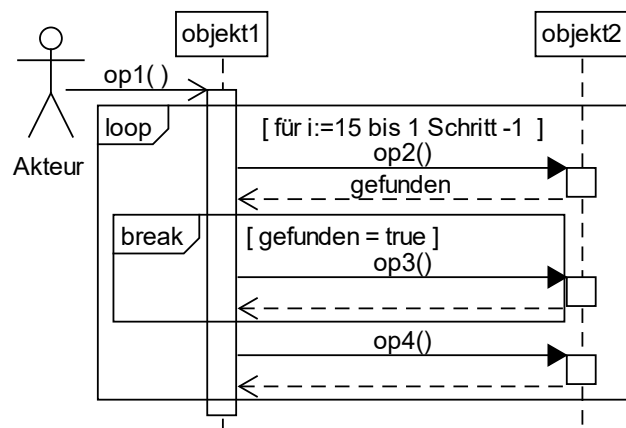
### Alternative – mehrseitige Verzweigung



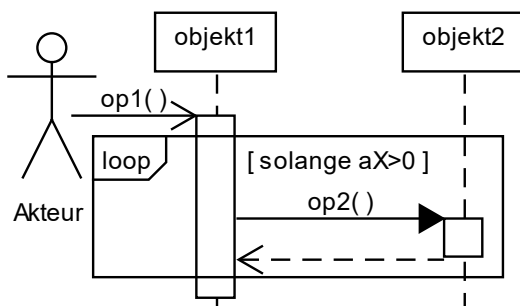
### Zählschleife



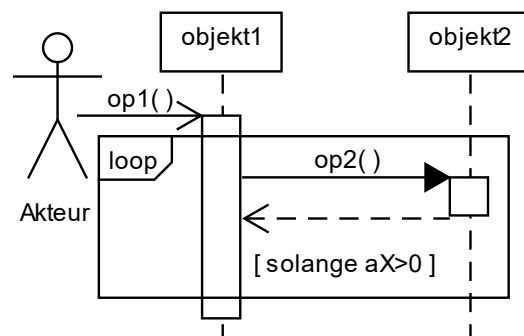
### Schleife mit Abbruch



### Kopfgesteuerte Schleife

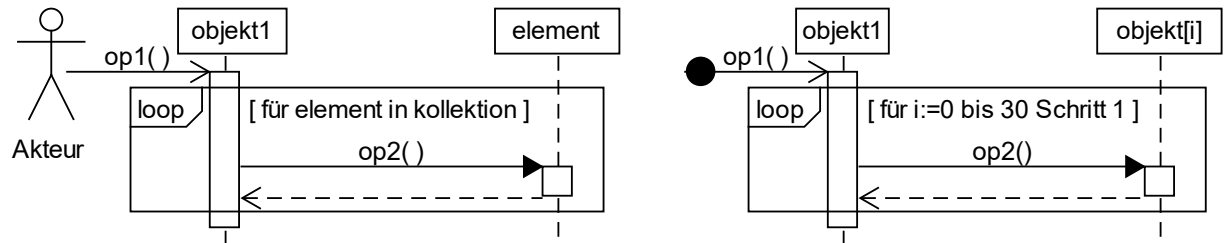


### Fußgesteuerte Schleife



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### Schleife über Kollektion



● op1() → Nachricht, bei welcher der Sender nicht spezifiziert ist.

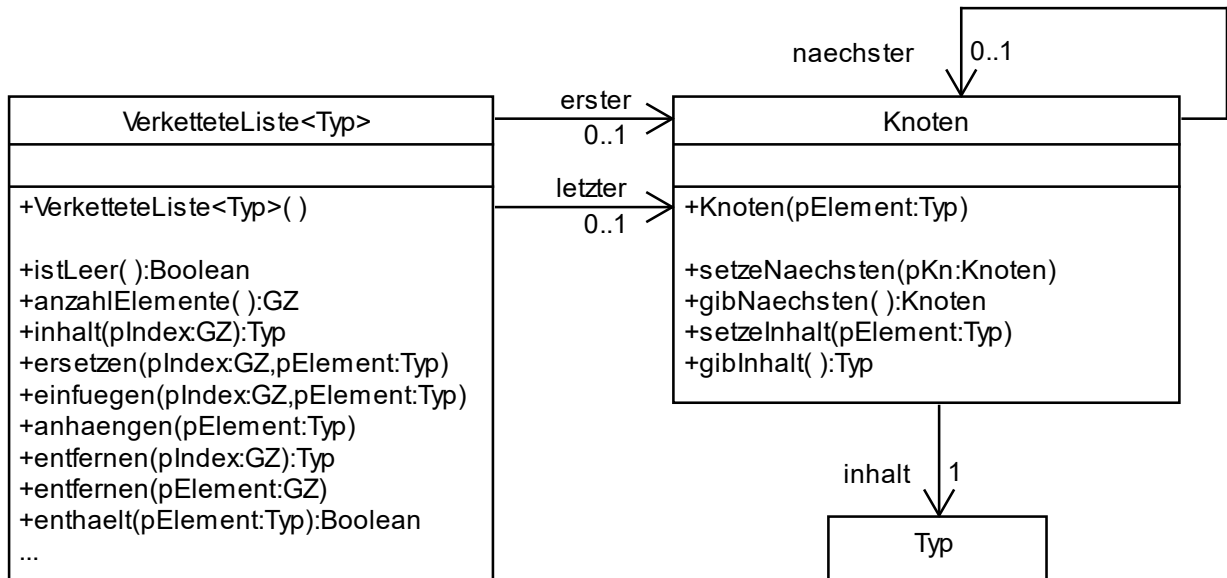
## 4.9 Zustandsdiagramme

Zustandsdiagramme siehe Kapitel 1

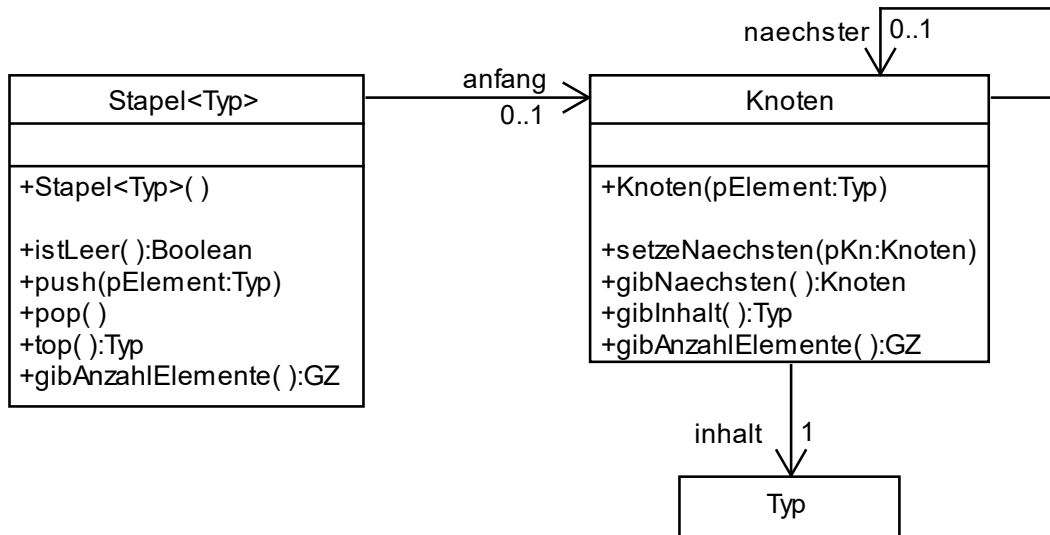
|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 5 Datenstrukturen

### 5.1 Verkettete Liste

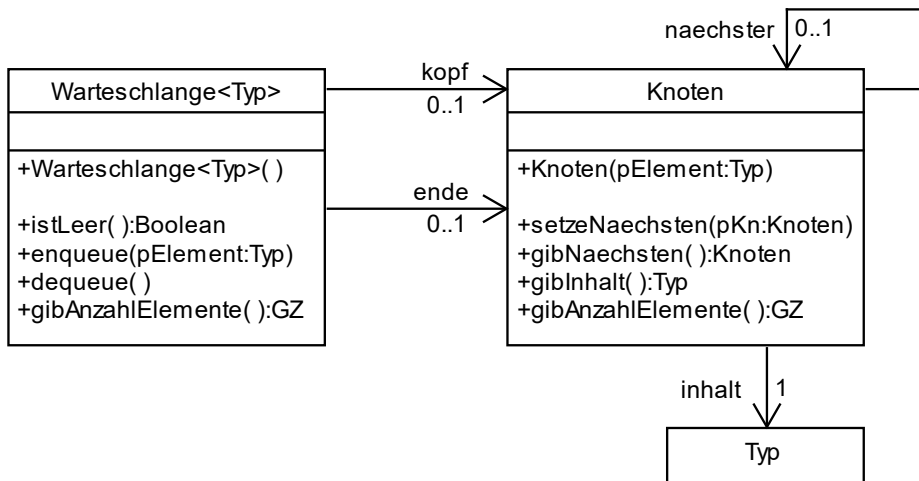


### 5.2 Stapel



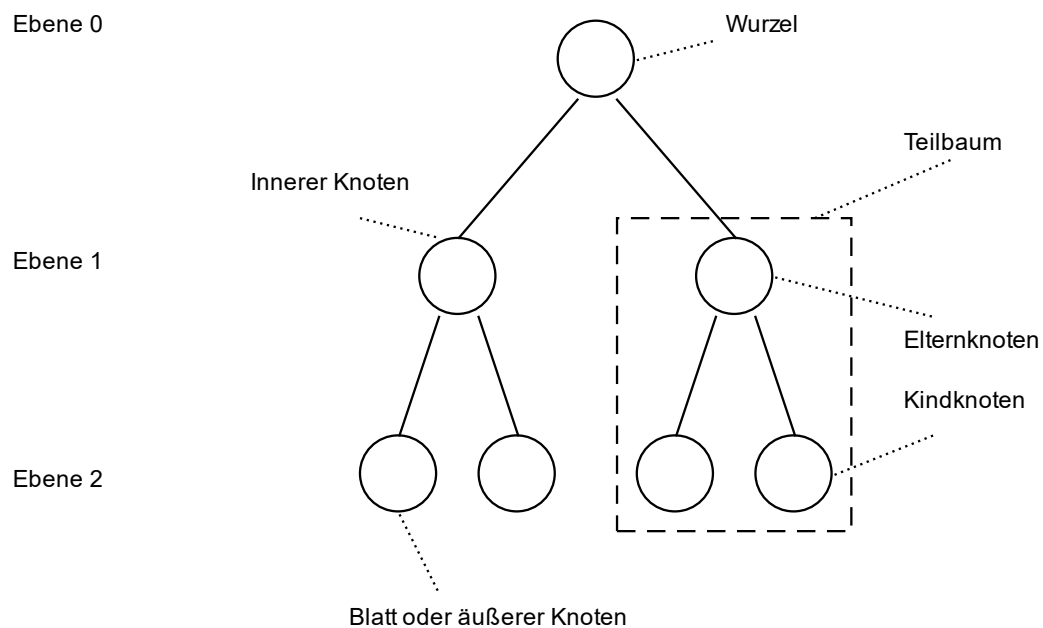
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 5.3 Warteschlange



### 5.4 Binärbaum

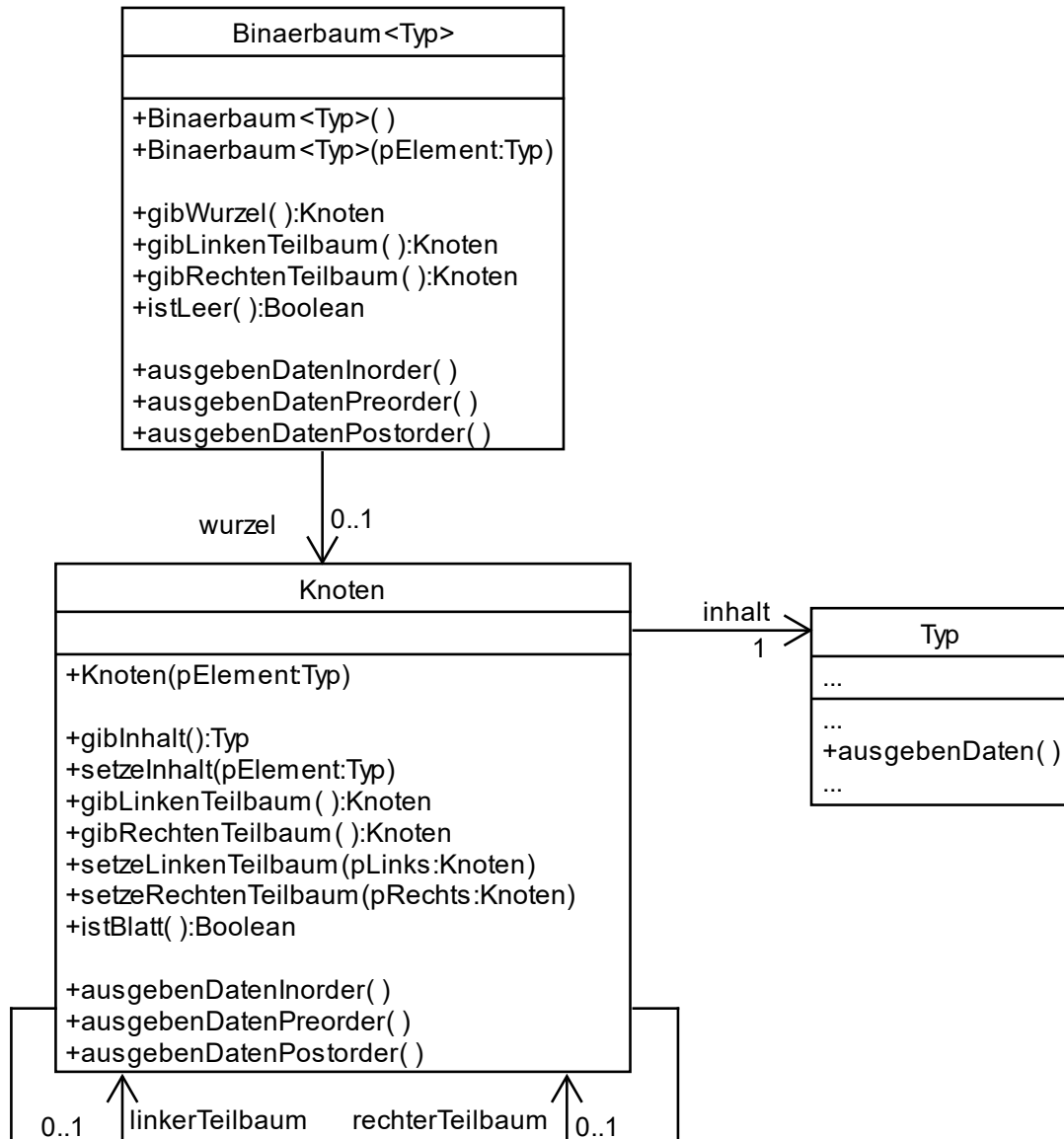
#### 5.4.1 Beispiel für einen Binärbaum der Tiefe 3





|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

### 5.4.2 Datenstruktur



### 5.4.3 Operation **ausgebenDatenInorder()** der Klasse **Knoten** in Pseudocode

OPERATION **ausgebenDatenInorder()** der Klasse **Knoten**

```

WENN linkerTeilbaum != NICHTS
    linkerTeilbaum.ausgebenDatenInorder()
ENDE WENN
inhalt.ausgebenDaten()
WENN rechterTeilbaum != NICHTS
    rechterTeilbaum.ausgebenDatenInorder()
ENDE WENN
    
```

|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 6 Künstliche Intelligenz

### 6.1 Klassifikation

Distanzfunktionen für  $x = (x_1, \dots, x_n)$  und  $y = (y_1, \dots, y_n)$

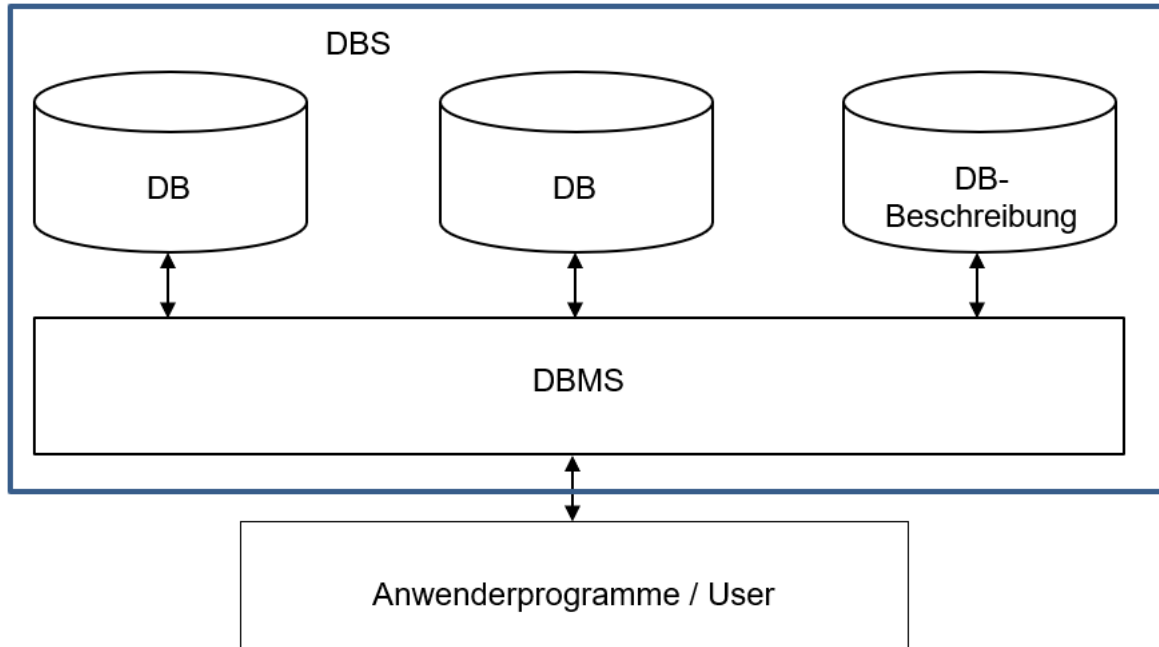
- Euklidische Distanz  $d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
- Manhattan-Distanz  $d(x, y) = \sum_{i=1}^n |x_i - y_i|$
- Maximum-Distanz  $d(x, y) = \max(|x_i - y_i|)$

*Anmerkung: Mit der Erweiterung des KI-Themenumfangs in zukünftigen Abiturprüfungen durch Anforderungserlässe wird auch in den nächsten Jahren die Formelsammlung im Bereich Künstliche Intelligenz erweitert.*

|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 7 Datenbanken

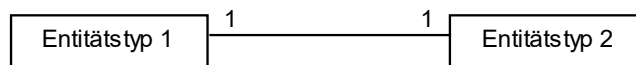
### 7.1 Datenbankmanagementsystem



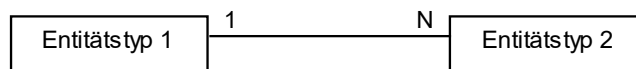
DBS = Datenbanksystem    DBMS = Datenbankmanagementsystem    DB = Datenbank

### 7.2 Entity-Relationship-Diagramm (ER-Diagramm)

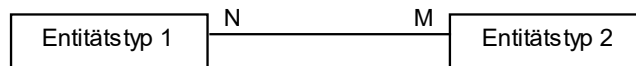
1:1 Beziehung



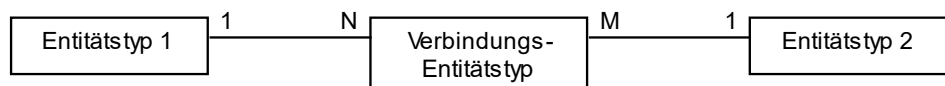
1:N Beziehung



N:M Beziehung



N:M Beziehung  
aufgelöst



|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

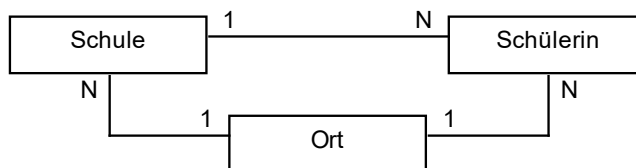
### 7.3 Relationenmodell

Alle Entitätstypen des Entity-Relationship-Diagramms mit Primär- und Fremdschlüsseln und allen Attributen der Entitätstypen in folgender Form:

Entitätstyp(Primärschlüssel, Attribut1, Attribut2, ..., Fremdschlüssel1, ...)

*Beispiel: Schülerinnen, die ein Mädchengymnasium besuchen.*

#### Entity-Relationship-Diagramm



#### Relationenmodell

Ort(OrtsNr, PLZ, Name)

Schule(SchulNr, Schulname, Straße, OrtsNr)

Schülerin(SchuelerinNr, Vorname, Name, Straße, OrtsNr, SchulNr)

### 7.4 Abfrageformulierung mit SQL

#### 7.4.1 Projektion und Formatierung

Auswahl aller Spalten einer Tabelle

Syntax:       SELECT       \*  
              FROM        <Tabelle>;

Auswahl mehrerer Spalten einer Tabelle

Syntax:       SELECT       <Spalte1>,<Spalte2>,<Spalte3>  
              FROM        <Tabelle>;

Auswahl ohne mehrfaches Auftreten derselben Zeile

Syntax:       SELECT DISTINCT   <Spalte>  
              FROM        <Tabelle>;

Umbenennen von Spalten bei der Ausgabe

Syntax:       SELECT       <Spalte> AS <neuer Spaltenname>  
              FROM        <Tabelle>;

Sortierung aufsteigend (ASC (optional)) oder absteigend (DESC)

Syntax:       SELECT       <Spalte>  
              FROM        <Tabelle>  
              ORDER BY    <Spalte> [ASC];  
  
              SELECT       <Spalte>  
              FROM        <Tabelle>  
              ORDER BY    <Spalte> DESC;

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

**Beispiel** Relationenmodell Schüler (SID, Vorname, Name, Klasse)

```
SELECT *
FROM Schüler
ORDER BY Name, Vorname;

SELECT Vorname, Name
FROM Schüler
ORDER BY Name ASC;

SELECT DISTINCT Klasse
FROM Schüler
ORDER BY Klasse DESC;

SELECT Name AS "Nachname", Vorname
FROM Schüler;
```

## 7.4.2 Selektion

Auswahl von Zeilen

Syntax:       SELECT     <Spalte>  
              FROM       <Tabelle>  
              WHERE       <Bedingung>;

Vergleichsoperatoren     =, <>, >, <, >=, <=               ( <> ungleich)  
                          BETWEEN wert1 AND wert2  
                          LIKE '\_...%' oder "\_...%"           ( \_ ein Zeichen  
  % beliebig viele Zeichen)  
                          IN ('Wert1','Wert2') oder       IN ("Wert1","Wert2")  
                          NOT IN ('Wert1','Wert2','Wert3')  
                          IS NULL  
                          IS NOT NULL

Logische Operatoren     AND, OR, NOT

**Beispiel** Relationenmodell Schüler (SID, Vorname, Name, Klasse)

```
SELECT *
FROM Schüler

Alle Schüler der TGI-J2
WHERE Klasse = "TGI-J2";

Alle Schüler der TG-Klassen
WHERE Klasse LIKE 'TG%';

Alle Schüler der TGI-Klassen
WHERE Klasse IN ('TGI-E', 'TGI-J1', 'TGI-J2');

Alle Schüler, die noch keiner Klasse zugeordnet sind
WHERE Klasse IS NULL;
```

**Beispiel** Relationenmodell Laborübung(LID, Thema, Dauer)

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

```
SELECT *
FROM Laborübung

    Alle Laborübungen die mindestens 60 und höchstens 90 Minuten ( $60 \leq \text{Dauer} \leq 90$ ) gedauert
    haben
    WHERE Dauer BETWEEN 60 AND 90;

    Alle Laborübungen, deren Themen nichts mit Radioaktivität oder Atmosphärenchemie zu tun
    haben
    WHERE Thema NOT IN ("Radioaktivität", "Atmosphärenchemie");

    Alle Laborübungen zur Organik, die kürzer als 60 Minuten waren
    WHERE Thema = "Organik"
        AND Dauer < 60;
```

### 7.4.3 Verbund von Tabellen

#### Equi-Join

Syntax:        SELECT     <Spalte1>,<Spalte2>  
              FROM        <Tabelle1>,<Tabelle2>  
              WHERE        <Join-Bedingung>;

In der Join-Bedingung wird festgelegt, dass der Inhalt bestimmter Spalten identisch sein muss.

**Beispiel** Relationenmodell     Schüler (SID, Vorname, Name, Klasse)  
                                 Teilnahme(TID, SID, LID, Datum, Punkte)  
                                 Laborübung(LID, Thema, Dauer)

```
SELECT Vorname, Name, Datum, Punkte
FROM    Schüler, Teilnahme
WHERE   Schüler.SID = Teilnahme.SID;
```

Anmerkung: Tabellennamen können in FROM durch Aliase abgekürzt werden.

```
SELECT Vorname, Name, Datum, Punkte
FROM    Schüler S, Teilnahme T
WHERE   S.SID = T.SID;
```

```
SELECT Vorname, Name, Datum, Thema, Dauer
FROM    Schüler S, Teilnahme T, Laborübung L
WHERE   S.SID = T.SID
        AND   L.LID = T.LID;
```

#### Inner Join mit zwei Tabellen

Syntax:        SELECT     A.<Spalte1>,B.<Spalte2>  
              FROM        <Tabelle1> A INNER JOIN <Tabelle2> B  
                             ON A.<Spalte1> = B.<Spalte2>

**Beispiel** Relationenmodell     Schüler (SID, Vorname, Name, Klasse)  
                                 Teilnahme(TID, SID, Datum, Punkte)

```
SELECT Vorname, Name, Datum, Punkte
FROM    Schüler S INNER JOIN Teilnahme T ON S.SID = T.SID;
```

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

#### 7.4.4 Aggregatfunktion

Aggregatfunktionen können auf einer ganzen Tabelle bzw. Zwischentabelle ausgeführt werden. Ihre Ergebnistabelle besteht dann aus einer Zelle.

Syntax:       SELECT     Aggregatfunktion(<Spalte>)  
              FROM       <Tabelle>;

|       |  |
|-------|--|
| SUM   | Summierung der numerischen Werte in der Spalte   |
| MIN   | Minimum der Spalte                               |
| MAX   | Maximum der Spalte                               |
| AVG   | Durchschnitt der numerischen Werte in der Spalte |
| COUNT | Anzahl der Zeilen des Zwischenergebnisses        |

Hinweis: NULL-Werte werden vor der Auswertung einer Aggregatfunktion eliminiert.

**Beispiel** Relationenmodell     Schüler (SID, Vorname, Name, Klasse)  
                                  Teilnahme(TID, SID, Datum, Punkte)

Summe der von den Schülern der Klasse TGI-E am 24.07.2021 erreichten Punkte  
SELECT SUM(Punkte) AS "Gesamtpunktzahl der Klasse TGI-E am 24.07.21"  
FROM   Schüler S, Teilnahme T  
WHERE   S.SID = T.SID  
      AND Klasse = "TGI-E"  
      AND Datum = #24/07/2021#;

Maximal erreichte Punktezahl  
SELECT MAX(Punkte) AS "Max. Punkte"  
FROM   Teilnahme;

Datum der ersten Teilnahme, d.h. des ersten Termins der Veranstaltung  
SELECT MIN(Datum) AS "Startdatum"  
FROM   Teilnahme;

Punktedurchschnitt der Klasse TGI-E  
SELECT AVG(Punkte) AS "Klassendurchschnitt TGI-E"  
FROM   Schüler S, Teilnahme T  
WHERE   S.SID = T.SID  
      AND Klasse = "TGI-E";

Anzahl der Schüler in der Klasse TGI-E  
SELECT COUNT(\*) AS "Anzahl Schüler TGI-E"  
FROM   Schüler  
WHERE   Klasse = "TGI-E";

Spezialfall: COUNT(DISTINCT ...)

**Beispiel** Relationenmodell     Schüler (SID, Vorname, Name, Klasse)  
                                  Teilnahme(TID, SID, Datum, Punkte)

Anzahl Klassen  
SELECT COUNT(DISTINCT Klasse) AS "Anzahl Klassen"  
FROM   Schüler S, Teilnahme T  
WHERE   S.SID = T.SID;

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 7.4.5 Aggregatfunktion mit Gruppierung

Mit GROUP BY werden Abfrageergebnisse nach bestimmten Kriterien in Gruppen zusammengefasst. Auf jeder Gruppe wird einzeln die Aggregatfunktion ausgewertet und ein eigener Wert berechnet. Somit besteht die Ergebnistabelle aus den Aggregatwerten der einzelnen Gruppen.

Syntax:        SELECT        <Spalte1>, Aggregatfunktion(<Spalte2>) AS <Name>  
              FROM        <Tabelle>  
              GROUP BY    <Spalte1>;

**Beispiel** Relationenmodell    Schüler (SID, Vorname, Name, Klasse)  
                                 Teilnahme(TID, SID, Datum, Punkte)

Punktedurchschnitte pro Klasse  
SELECT    Klasse, AVG(Punkte) AS "Gesamtpunktzahl pro Klasse"  
FROM      Schüler S, Teilnahme T  
WHERE     S.SID = T.SID  
GROUP BY Klasse;

Beste Leistung pro Tag  
SELECT    Datum, MAX(Punkte) AS "Bestes Tagesergebnis"  
FROM      Teilnahme T  
GROUP BY Datum;

### 7.4.6 Selektion von Gruppen

Im Unterschied zur einfachen Selektion mit SELECT können mit HAVING Abfrageergebnisse von Aggregatfunktionen auf Gruppen selektiert werden.

Syntax:        SELECT        <Spalte1>, Aggregatfunktion(<Spalte2>) AS <Name>  
              FROM        <Tabelle>  
              WHERE        <Bedingung>  
              GROUP BY    <Spalte1>  
              HAVING        <Bedingung für Aggregatfunktion>;

**Beispiel** Relationenmodell    Schüler (SID, Vorname, Name, Klasse)  
                                 Teilnahme(TID, SID, Datum, Punkte)

Punktedurchschnitte pro Klasse, aber nur wenn der Durchschnitt größer als 20 Punkte ist.  
SELECT    Klasse, AVG(Punkte) AS "Gesamtpunktzahl pro Klasse"  
FROM      Schüler S, Teilnahme T  
WHERE     S.SID = T.SID  
GROUP BY Klasse  
HAVING    AVG(Punkte)>20;

### 7.4.7 Komplette SQL-Anweisung

Syntax:        SELECT        ...  
              FROM        ...  
              WHERE        ...  
              GROUP BY    ...  
              HAVING        ...  
              ORDER BY    ... ;

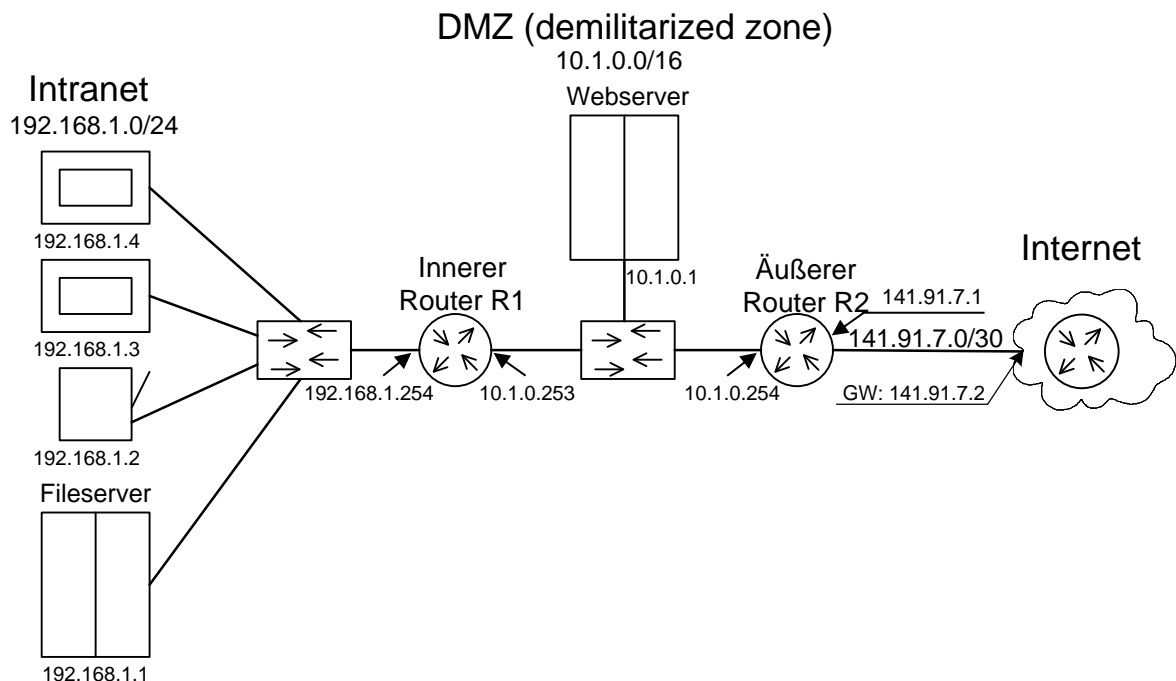
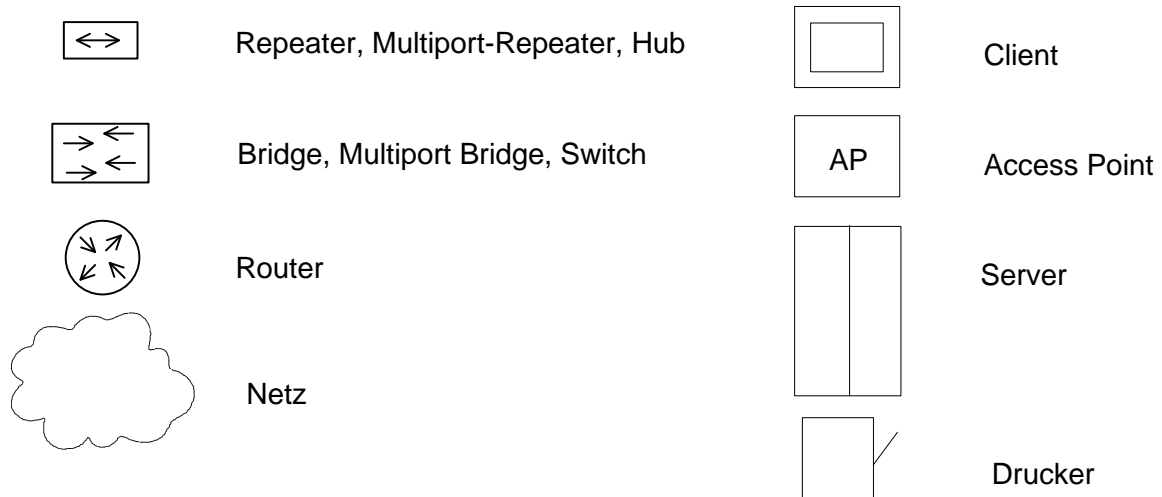


|                       |                            |
|-----------------------|----------------------------|
| Abiturprüfung ab 2024 | Berufliches Gymnasium (TG) |
| Formelsammlung        | 1.5.2 Informationstechnik  |

## 8 Vernetzte Systeme

### 8.1 Netzwerktechnik

#### 8.1.1 Netzwerksymbole



#### 8.1.2 Routing-Tabelle (IPv4)

Die Routingtabelle des Router R2 sieht folgendermaßen aus:

| Netzadresse | Subnetzmaske | Gateway    |
|-------------|--------------|------------|
| 141.91.7.0  | /30          | *          |
| 10.1.0.0    | /16          | *          |
| 192.168.1.0 | /24          | 10.1.0.253 |
| 0.0.0.0     | 0.0.0.0      | 141.91.7.2 |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

### 8.1.3 Aufbau IPv4-Adresse

IP-Adresse (dotted-decimal-format): z.B.

IP-Adresse (binär):

177 . 17 . 223 . 1  
 10110001.00010001.11011111.00000001  
 8 Bit = 1 Oktett  
 32 Bit = 4 Bytes

IP-Adresse z.B. 192.168. 1 . 1 → 11000000.10101000.00000001.00000001  
 Netzmaske z.B. /24 = 255.255.255. 0 → 11111111.11111111.11111111.00000000  
 Netz-ID 192.168. 1 . 0 ← 11000000.10101000.00000001.00000000  
 Host-ID 0 . 0 . 0 . 1 ← 00000000.00000000.00000000.00000001

Alle Host-ID-Bits = 0: Netz-Adresse, hier 192.168.1.0

Alle Host-ID-Bits = 1: Broadcast-Adresse, hier 192.168.1.255

### 8.1.4 Aufbau IPv6-Adresse

IP-Adresse (hexadezimal): z.B.

2001:07C0:8280:0253:0000:0000:0000:0020  
 16 Bit  
 8 Blöcke (16 Bit) = 128 Bit

Weitere IPv6-Schreibweise:

Führende Nullen können ausgelassen werden → 2001:7C0:8280:253:0:0:0:20

Aufeinanderfolgende Null-Blöcke können durch zwei Doppelpunkte einmal ersetzt werden → 2001:7C0:8280:253::20

IPv4-Adressen können in IPv6-Adressen eingebettet werden, z.B. 192.168.1.1 → 0:0:0:0:0:0:192.168.1.1  
 → ::192.168.1.1 → ::C0A80101

Adressformat:

|                       |            |                            |  |
|-----------------------|------------|----------------------------|--|
| 64 Bits               |            | 64 Bits                    |  |
| Netzwerk Präfix       |            | Interface Identifier (IID) |  |
| 48 Bits               | 16 Bits    |                            |  |
| Global Routing Präfix | Subnetz ID |                            |  |

Netzwerk-Präfix:

**2001:07C0:8280:0253** → Global Routing Präfix

2001:07C0:8280:**0253** → Subnetz Identifier

Adressbereich-Zuweisung:

2001:07C0:8280:0253::**64**

2001:07C0:8280:0200::**56**

2001:07C0:8280::**48**

Interface Identifier:

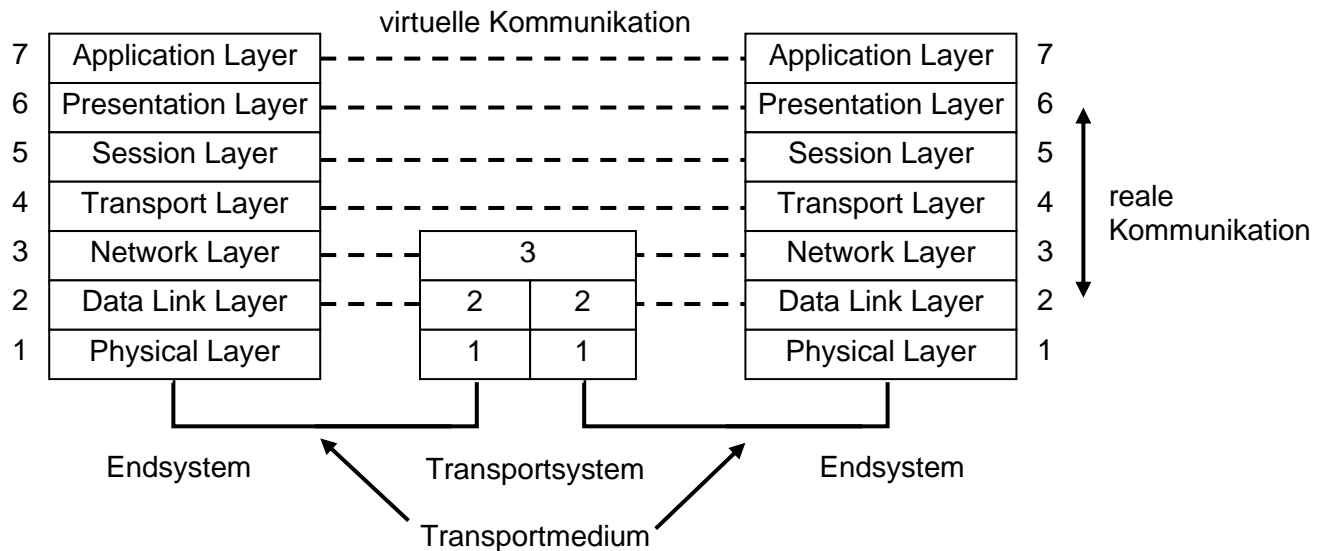
xxxx:xxxx:xxxx:xxxx:**0000:0000:0000:0020**

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

2001:07C0::/32

## 8.2 Schichtenmodelle

### 8.2.1 ISO-OSI-7-Schichtenmodell



### 8.2.2 TCP-IP-Schichtenmodell

| OSI-Schicht | TCP/IP-Schicht | Protokoll-Beispiele                      |
|-------------|----------------|--|
| 7           | Anwendungen    | HTTP, FTP, SMTP, Telnet, DHCP, MQTT, ... |
| 6           |                |  |
| 5           |                |  |
| 4           | Transport      | TLS                                      |
| 3           | Internet       | TCP, UDP                                 |
| 2           | Netzzugang     | IP (IPv4, IPv6), ICMP                    |
| 1           |                | Ethernet                                 |

## 8.3 Header

### 8.3.1 Ethernet II

| Präambel | Zieladresse | Absenderadresse | Typ | Daten     | Link Trailer |
|----------|-------------|-----------------|-----|-----------|--------------|
| 8        | 6           | 6               | 2   | 46...1500 | 4 Byte       |

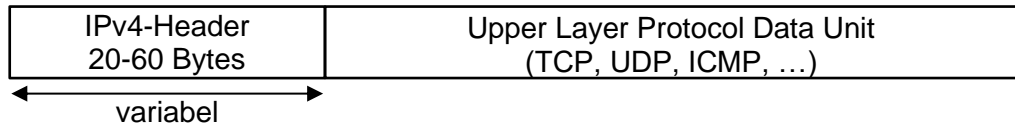
### 8.3.2 IPv4-Header

| Byte  | Inhalt                  |
|-------|-------------------------|
| 0     | Version   IHL           |
| 1     | TOS                     |
| 2-3   | Paketlänge              |
| 4-5   | Identifikation          |
| 6     | Flags   Fragmentabstand |
| 7     | Fragmentabstand         |
| 8     | Time To Live (TTL)      |
| 9     | Protokoll               |
| 10-11 | Kopf-Prüfsumme          |
| 12-15 | IP-Sendeadresse         |
| 16-19 | IP-Empfängeradresse     |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

|        |                                  |
|--------|----------------------------------|
| 20 ... | Optionen (mit evtl. Füllzeichen) |
|--------|----------------------------------|

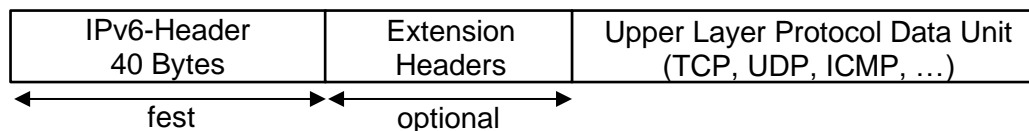
IPv4-Paketstruktur:



### 8.3.3 IPv6-Header

| Byte  | Inhalt                                   |
|-------|--|
| 0-3   | Version   Traffic Class   Flow Label     |
| 4-7   | Payload Length   Next Header   Hop Limit |
| 8-23  | Source Address                           |
| 24-39 | Destination Address                      |

IPv6-Paketstruktur:



### 8.3.4 TCP –Header

| Byte   | Inhalt  |
|--------|---|
| 0-1    | Source Port                                       |
| 2-3    | Destination Port                                  |
| 4-7    | Sequenznummer                                     |
| 8-11   | Quittungsfeld (Piggyback, Acknowledgement Number) |
| 12     | Header-Länge   reserviert                         |
| 13     | reserviert   URG   ACK   PSH   RST   SYN   FIN    |
| 14-15  | Fenster Größe                                     |
| 16-17  | Prüfsumme   |
| 18-19  | Urgent Zeiger                                     |
| 20 ... | Optionen (evtl. mit Füllzeichen)                  |

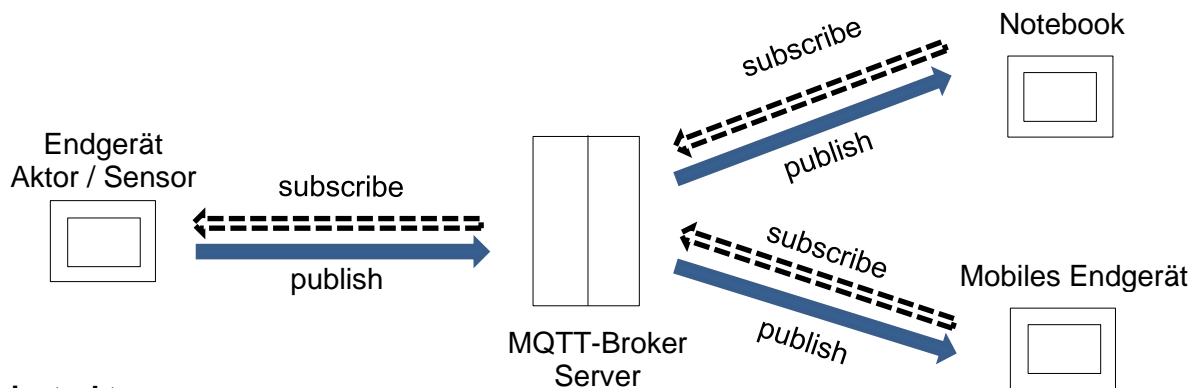
### 8.3.5 UDP –Header

| Byte | Inhalt               |
|------|----------------------|
| 0-1  | Source Port          |
| 2-3  | Destination Port     |
| 4-5  | Länge des Datagramms |
| 6-7  | Check-Summe          |

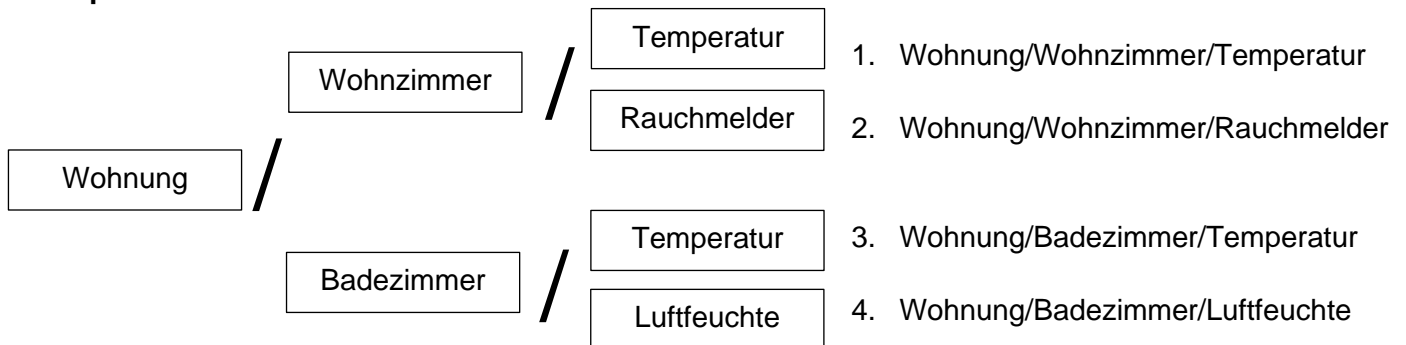
|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 8.4 Internet der Dinge (IoT)

### 8.4.1 MQTT-Protokoll (Message Queuing Telemetry Transport)



#### Topicstruktur:



#### Multi-Level-Wildcard: #

Wohnung/Wohnzimmer/#

#### Single-Level-Wildcard: +

Wohnung+/Temperatur

#### Qualitätsstandards:

QoS 0

publish

QoS 1

publish

puback

QoS 2

publish

pubrec

pubrel

pubcomp

#### Ports:

1883 : MQTT, unverschlüsselt

8883 : MQTT, verschlüsselt

8884 : MQTT, verschlüsselt, Client Zertifikat notwendig

8080 : MQTT über WebSockets, unverschlüsselt

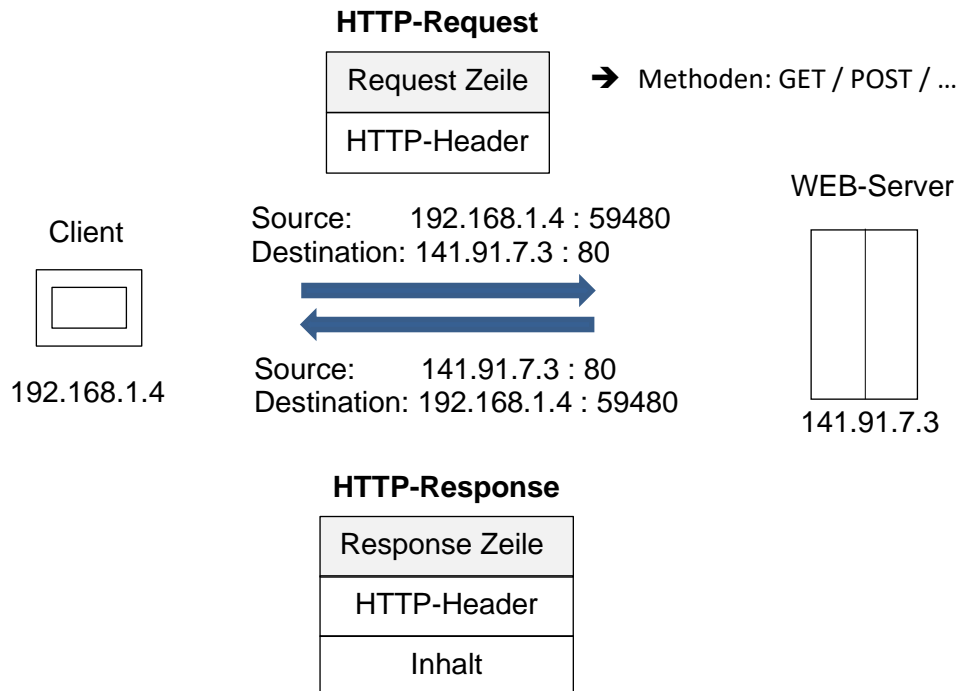
#### MQTT -Header: (Beispiel - Publish Message)

| Byte | Inhalt   |
|------|--|
| 0    | Nachrichtentyp (4 Bit)   Dub-Flag   Quality of Service   Retain-Flag |
| 1    | Länge des restlichen MQTT-Pakets                                     |
| ...  | MQTT-Topic → Topic-Länge / Topic / Payload                           |

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

## 8.4.2 HTTP-Protokoll (Hypertext Transfer Protocol)

**Kommunikationsprinzip:**



**URL (Uniform Resource Locator):**

| Protokoll | Domain  | Pfad                                     |
|-----------|---------|--|
| https://  | gsoe.de | /bildungsangebote/technisches-gymnasium/ |

**Ports:**

**80** : HTTP, unverschlüsselt

**443** : HTTPS, verschlüsselt

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Abiturprüfung ab 2024</b> | <b>Berufliches Gymnasium (TG)</b> |
| <b>Formelsammlung</b>        | <b>1.5.2 Informationstechnik</b>  |

#### Request HTTP 1/1

| Method | Path   | Protocol     |
|--------|--|--------------|
| GET    | /wp/content/uploads/2020/11/pixels-fauxels.jpg | HTTP/1.1\r\n |

| HTTP-Header - Name: Wert (Beispiele) |   |
|--------------------------------------|---|
| Host:                                | → Domain-Name des Servers   |
| User-Agent:                          | → User-Agent des Clients  |
| Accept:                              | → Welche Inhaltstypen der Client verarbeiten kann   |
| z.B.                                 | <ul style="list-style-type: none"> <li>Accept-Charset: → Welche Zeichensätze der Client anzeigen kann.</li> <li>Accept-Encoding: → Welche komprimierten Formate der Client unterstützt.</li> <li>Accept-Language: → Gewünschte Sprachversion</li> </ul> |
| Date:                                | → Datum und Zeit des Requests   |
| Connection:                          | → Bevorzugte Art der Verbindung   |
| Referrer:                            | → URL der Ressource, von der aus verlinkt wurde.  |
| Content-Length:                      | → Länge des Request-Bodys   |
| Content-Type:                        | → MIME-Typ des Bodys (bei POST- und PUT-Requests)   |

#### Response HTTP 1/1

| Protocol | Status-Code |
|----------|-------------|
| HTTP/1.1 | 200 OK\r\n  |

| HTTP-Header - Name: Wert (Beispiele) |  |
|--------------------------------------|--|
| Date:                                | → Zeitpunkt der Response                 |
| Server:                              | → Kennung des Servers                    |
| Accept-Ranges:                       | → Welche Einheiten der Server akzeptiert |
| Allow:                               | → Erlaubte Request-Typen (Methoden)      |
| Connection:                          | → Bevorzugte Art der Verbindung          |

| Status-Codes | (Beispiele)  |
|--------------|--|
| 100 ... 199: | Information  |
| 200 ... 299: | Client-Anfrage erfolgreich<br>z.B. <b>200</b> – OK   |
| 300 ... 399: | Client-Anfrage umgeleitet<br>z.B. <b>301</b> – Moved Permanently<br><b>302</b> – Moved Temporarily |
| 400 ... 499: | Fehlen des Dokuments<br>z.B. <b>403</b> – Forbidden<br><b>404</b> – Not Found                      |
| 500 ... 599: | Serverfehler   |