

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Formelsammlung

1.5.2 TG Informationstechnik

Formelsammlung Allgemein

Version: V 4.57

Gültig ab Abitur 2024

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Inhaltsverzeichnis:

1	Beschreibung von Systemzuständen mit UML-Zustandsdiagrammen.....	4
1.1	UML-Zustandsdiagramme (allgemein)	4
	Varianten von Transitionen	5
1.2	Allgemeines für UML-Zustandsdiagramme.....	6
1.3	Ergänzungen für Mikrocontroller.....	6
	Zustandsdefinition in C/CPP.....	6
	Zustandsvariable C/CPP	6
	Der Start-Pseudozustand	7
	Verhalten.....	7
	Zustandsübergang mit Wächterbedingung	7
	Zustandsübergang mit Ereignis und Wächterbedingung	8
	Selbsttransition.....	8
2	Hardware - Digitaltechnik.....	9
2.1	Logikgatter	9
2.2	Schaltnetze	10
2.3	Schaltwerke	11
	Taktgenerator.....	11
	Flip-Flops	11
	RAM.....	11
	ROM	11
	Schieberegister	12
	Zähler (Blockschaltbild)	12
	Zähler (4-Bit)	12
2.4	Sensoren.....	12
2.5	Aktoren.....	13
3	Hardware - Mikrocontrollertechnik.....	13
3.1	Blockschaltbild „Prüfungscontroller“	13
3.2	Prozessorarchitektur	14
	Programmiermodell.....	14
	Blockbild Prozessorkern CPU	14
	Blockbild Mikrocontroller	14
	Blockschaltbild Mikrocontroller	15
	Befehlspipeline einer RISC-CPU.....	15
	Speicherarchitektur	15
4	Programmentwicklung und Objektorientierter Entwurf	16
4.1	Vergleichsoperatoren für Bedingungen (Pseudocode)	16
4.2	Kontrollstrukturen (Pseudocode)	16
4.3	Datentypen.....	17
	Elementare Datentypen.....	17
	Komplexe Datentypen	17
4.4	Klassen	19
	Attribute.....	19
	Operationen	20
	Assoziationen, Rollennamen und Multiplizitäten.....	21
4.5	Vererbung	21
4.6	Abstrakte Klassen und Schnittstellen	21
4.7	Objektdiagramme.....	22
4.8	Sequenzdiagramme	23
4.9	Zustandsdiagramme.....	25

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

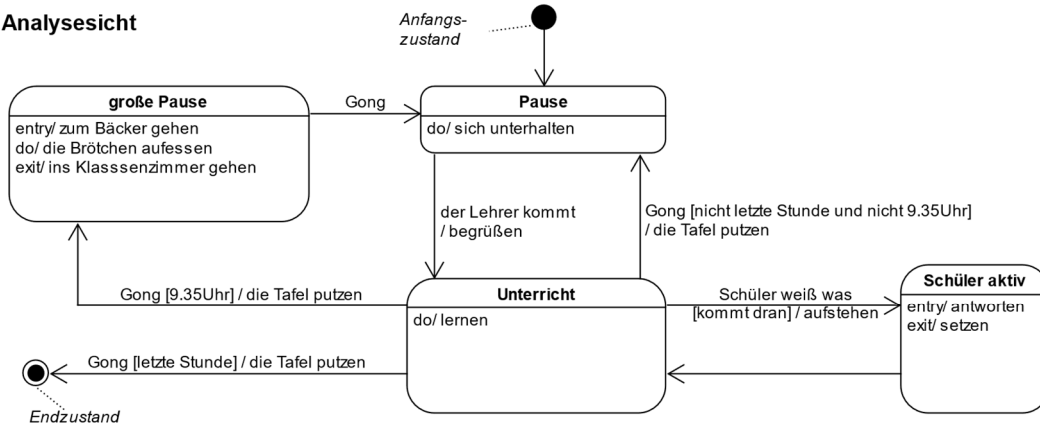
5	Datenstrukturen	26
5.1	Verkettete Liste	26
5.2	Stapel	26
5.3	Warteschlange	27
5.4	Binärbaum	27
	Beispiel für einen Binärbaum der Tiefe 3	27
	Datenstruktur	28
	Operation <code>ausgebenDatenInOrder()</code> der Klasse <code>Knoten</code> in Pseudocode	28
6	Künstliche Intelligenz	29
6.1	Klassifikation	29
6.2	Gini-Unreinheit	29
6.3	Normalisierung von Daten	29
7	Datenbanken	30
7.1	Datenbankmanagementsystem	30
7.2	Entity-Relationship-Diagramm (ER-Diagramm)	30
7.3	Relationenmodell	31
7.4	Abfrageformulierung mit SQL	31
	Projektion und Formatierung	31
	Selektion	32
	Verbund von Tabellen	33
	Aggregatfunktion	34
	Aggregatfunktion mit Gruppierung	35
	Selektion von Gruppen	35
	Komplette SQL-Anweisung	35
8	Vernetzte Systeme	36
8.1	Netzwerktechnik	36
	Netzwerksymbole	36
	Routing-Tabelle (IPv4)	36
	Aufbau IPv4-Adresse	37
	Aufbau IPv6-Adresse	37
8.2	Schichtenmodelle	38
	ISO-OSI-7-Schichtenmodell	38
	TCP-IP-Schichtenmodell	38
8.3	Header	38
	Ethernet II	38
	IPv4-Header	38
	IPv6-Header	39
	TCP-Header	39
	UDP-Header	39
	Firewall-Regelkatalog (vereinfacht)	40
8.4	Internet der Dinge (IoT)	41
	MQTT-Protokoll (Message Queuing Telemetry Transport)	41
	HTTP-Protokoll (Hypertext Transfer Protocol)	42

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

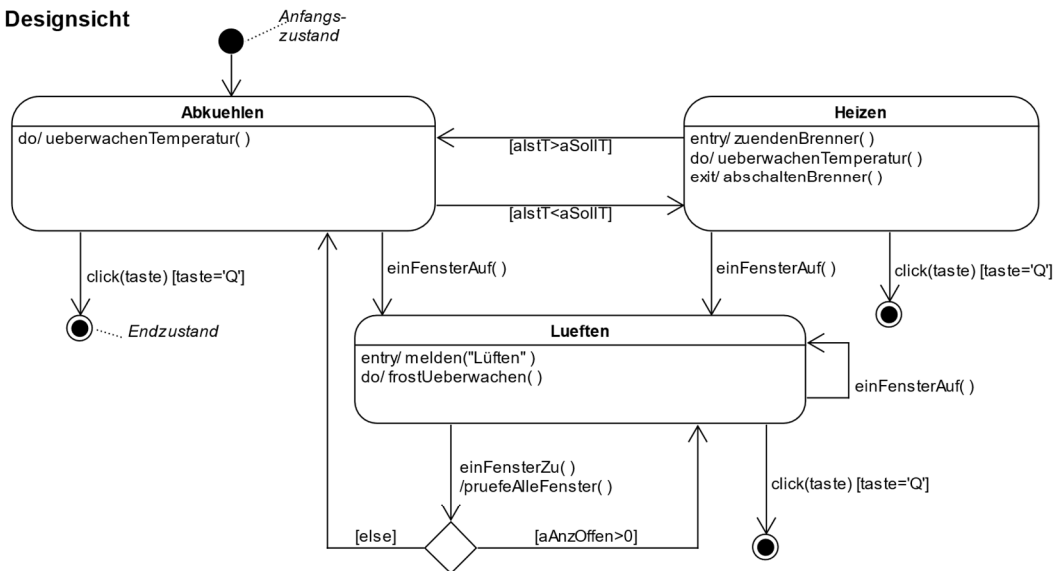
1 Beschreibung von Systemzuständen mit UML-Zustandsdiagrammen

1.1 UML-Zustandsdiagramme (allgemein)

Analysesicht



Designsicht

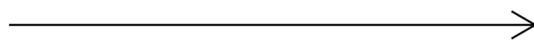


Notation	
Zustand	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p style="text-align: center;">Zustandsname</p> <p>entry/ Eintrittsverhalten do/ Andauerndes Verhalten exit/ Austrittsverhalten</p> </div>
Transitionen	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p style="text-align: center;">Zustandsname</p> </div>
Ereignisse	<p>Signal, Botschaft: Z.B. click(taste), einFensterAuf(), Gong</p> <p>Verlassen eines Zustanes ohne externes Ereignis z.B. beim Ende des do-Verhaltens (falls vorhanden)</p>
	<p>«implizit»</p> <p>/ Verhalten</p> <p>[Wächterbedingung]</p> <p>[Wächterbedingung] / Verhalten</p> <p>Ereignis</p> <p>Ereignis / Verhalten</p> <p>Ereignis [Wächterbedingung]</p> <p>Ereignis [Wächterbedingung] / Verhalten</p>

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

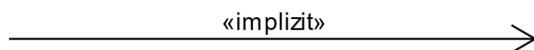
Varianten von Transitionen

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



a) Transition ohne Beschriftung:

Sie bewirkt einen unmittelbaren Zustandswechsel, nachdem das entry-Verhalten beendet wurde.

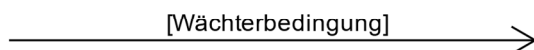


Alternative zu a)



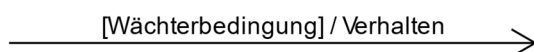
b) Transition mit Verhalten:

Verhalten wie bei a). Zusätzlich wird beim Zustandswechsel noch das Verhalten ausgeführt.



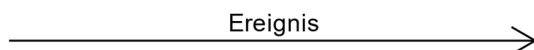
c) Transition mit Wächterbedingung:

Sie bewirkt einen Zustandswechsel, sobald die Wächterbedingung erfüllt ist.



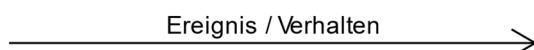
d) Transition mit Wächterbedingung und Verhalten:

Sie bewirkt einen Zustandswechsel, sobald die Wächterbedingung erfüllt ist. Beim Zustandswechsel wird das Verhalten ausgeführt.



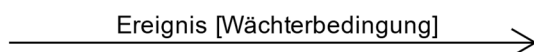
e) Transition mit Ereignis:

Sie bewirkt einen Zustandswechsel beim Eintreten des Ereignisses.



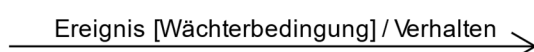
f) Transition mit Ereignis und Verhalten:

Verhalten wie bei d). Zusätzlich wird beim Zustandswechsel noch das Verhalten ausgeführt.



g) Transition mit Ereignis und Wächterbedingung:

Verhalten wie bei d), falls zusätzlich die Wächterbedingung erfüllt ist.



h) Transition mit Ereignis, Wächterbedingung und Verhalten

Verhalten wie bei f). Zusätzlich wird beim Zustandswechsel noch das Verhalten ausgeführt.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

1.2 Allgemeines für UML-Zustandsdiagramme

- ✓ ISR klein halten => Zustandsänderung mit Variable, Funktionsaufrufe wenn möglich vermeiden
- ✓ Funktionen = Operationen, wenn möglich mit Kleinbuchstaben beginnen
- ✓ `init()` bzw. `setup()` sollte als Verhalten nach Start beschrieben werden und nicht unbedingt als Anfangszustand
- ✓ Keine Unterscheidung UML-Ereignis und HW-Ereignis => somit `isr_name()` als Ereignis möglich
- ✓ `#define` bzw. `enum`-Werte mit Großbuchstaben
- ✓ Guard = Wächter - didaktische Interpretation => HW-IR-Einheit als „Aufpasser“ der eine Flagge zeigt und somit die Bedingung erfüllt für ein HW-Ereignis

1.3 Ergänzungen für Mikrocontroller

Hinweis: Die folgenden Codebeispiele sind nicht verbindlich

Zustandsdefinition in C/CPP

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

Allgemein	Beispiel
<code>#define Zustandsname Zustandsnummer</code>	<code>#define BLINKEN 1</code>
oder	
<code>enum zustandstyp {ZUSTANDSNAME=Zustandsnummer, ... }</code>	<code>enum zustandstyp {BLINKEN =1, ...};</code>

Zustandsvariable C/CPP

Ein Zustand kann durch eine Zustandsvariable gekennzeichnet werden:

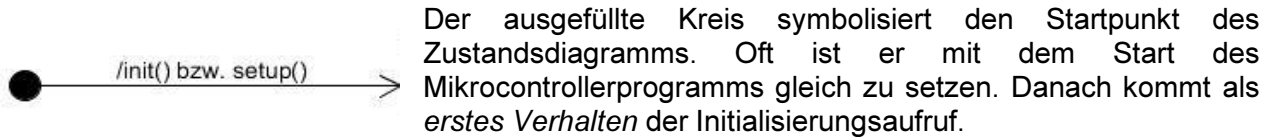
Beispiele	Erklärung
<code>int zustand;</code>	Zustandsvariable vom Typ <code>int</code>
<code>PortOut zustand(PortC,0xFF);</code>	Eine Portkonfiguration repräsentiert den Zustand
<code>zustandstyp zustand;</code>	Zustandsvariable als <code>enum</code> (siehe oben)

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.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

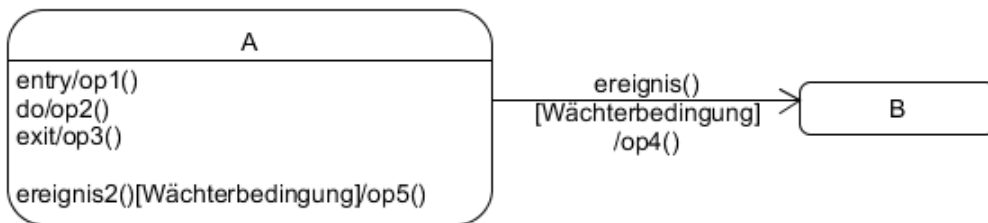
Der Start-Pseudozustand

Die meisten Zustandsdiagramme beginnen mit einem Start-Pseudozustand:



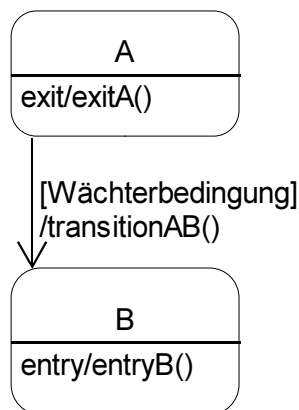
Verhalten

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()

Zustandsübergang mit Wächterbedingung



```

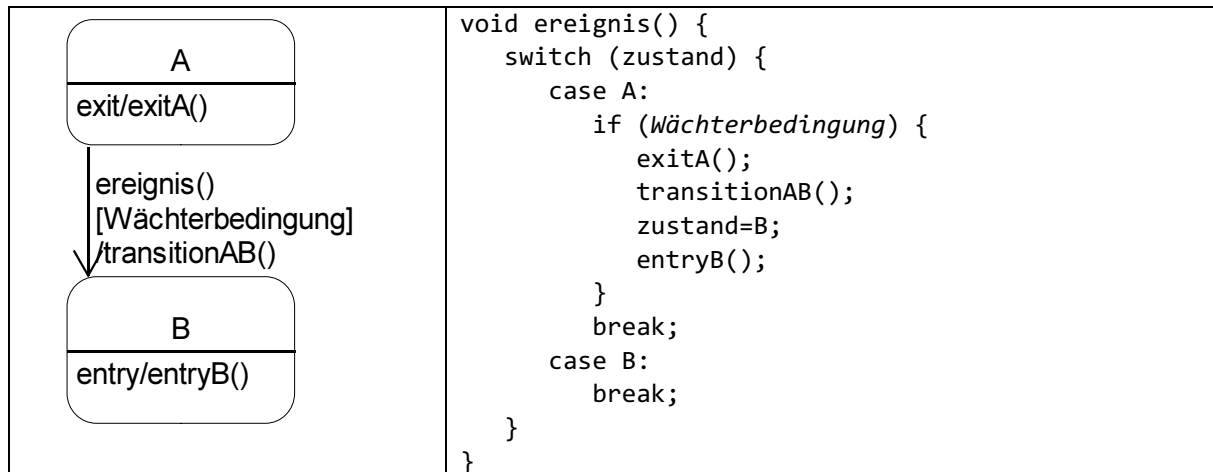
int main() {
    while(true) {
        switch (zustand) {
            case A:
                if (Wächterbedingung) {
                    exitA();
                    transitionAB();
                    zustand=B;
                    entryB();
                }
                break;
            case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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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()

Zustandsübergang mit Ereignis und Wächterbedingung

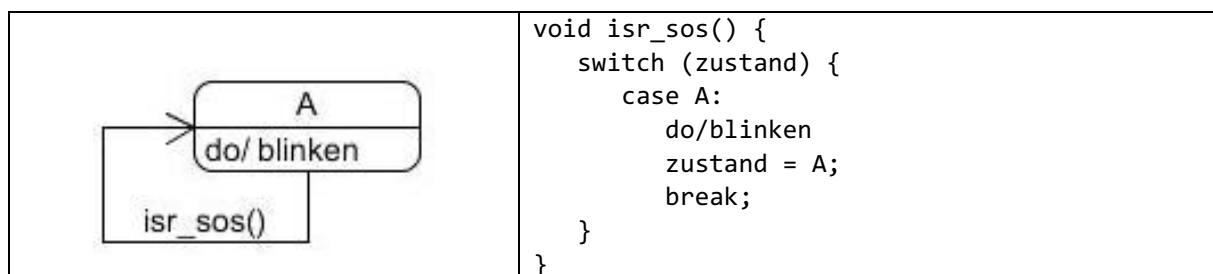


Es gibt **Aufruf-** und **Signal-Ereignisse**. Bei Signal-Ereignissen handelt es sich um Interrupts. Als Ereignisbezeichnung wird der Name der **Interrupt Service Routine** `isr_ereignis()` verwendet.

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()

Selbsttransition

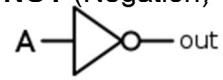
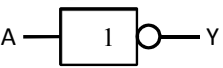
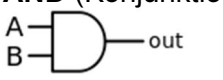
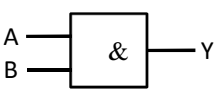
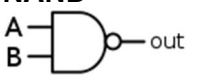
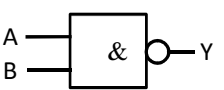
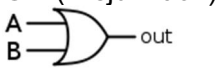
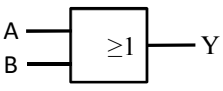
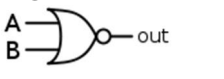
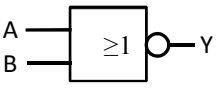
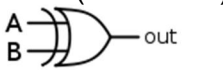
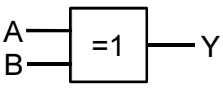
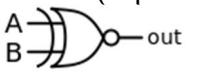
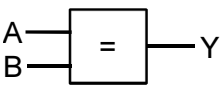


In der ISR `isr_sos()` wird zuerst der Zustand geprüft. Falls sich der Mikrocontroller im Zustand A befindet, wird die Wächterbedingung, falls vorhanden, an der Transition überprüft. Falls die Wächterbedingung erfüllt ist, erfolgt der Zustandswechsel wieder nach A.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

2 Hardware - Digitaltechnik

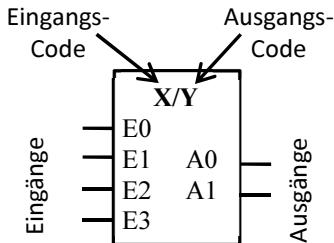
2.1 Logikgatter

<p>NOT (Negation)</p>   <table border="1" data-bbox="686 492 805 604"> <thead> <tr> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table> <p> $Y = \bar{A}$ $Y = !A$ $Y = \neg A$ </p>	A	Y	0	1	1	0																									
A	Y																														
0	1																														
1	0																														
<p>AND (Konjunktion)</p>   <table border="1" data-bbox="654 772 837 963"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = A \wedge B$ $Y = A \& B$ </p>	B	A	Y	0	0	0	0	1	0	1	0	0	1	1	1	<p>NAND</p>   <table border="1" data-bbox="1284 772 1468 963"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = \overline{A \wedge B}$ $Y = !(A \& B)$ </p>	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																													
<p>OR (Disjunktion)</p>   <table border="1" data-bbox="654 1086 837 1276"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = A \vee B$ $Y = A \# B$ </p>	B	A	Y	0	0	0	0	1	1	1	0	1	1	1	1	<p>NOR</p>   <table border="1" data-bbox="1284 1086 1468 1276"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = \overline{A \vee B}$ $Y = !(A \# B)$ </p>	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																													
<p>XOR (Antivalenz)</p>   <table border="1" data-bbox="662 1400 845 1590"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = (\bar{A} \wedge B) \vee (A \wedge \bar{B})$ $Y = (!A \& B) \# (A \& !B)$ </p>	B	A	Y	0	0	0	0	1	1	1	0	1	1	1	0	<p>XNOR (Äquivalenz)</p>   <table border="1" data-bbox="1292 1400 1476 1590"> <thead> <tr> <th>B</th> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <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> </tbody> </table> <p> $Y = (\bar{A} \wedge \bar{B}) \vee (A \wedge B)$ $Y = (!A \& !B) \# (A \& B)$ </p>	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																													

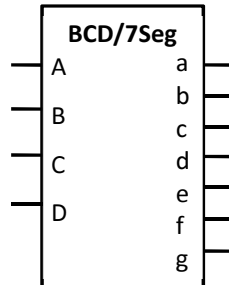
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

2.2 Schaltnetze

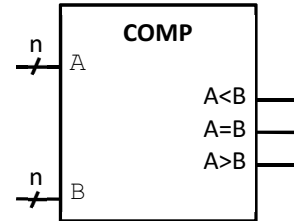
Codeumsetzer (Umcodierer)



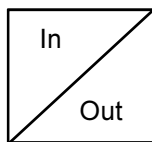
BCD zu 7 Seg



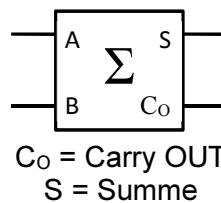
Vergleicher



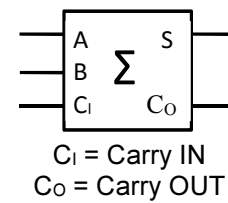
BSB Codewandler



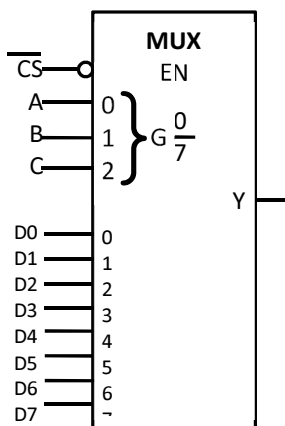
Halbaddierer



Volladdierer

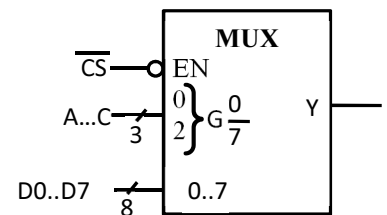


MUX (8 zu 1)



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

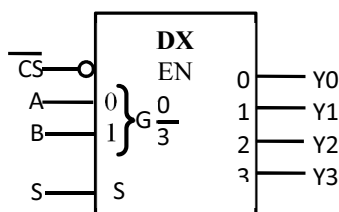
x = don't care



Adress- und Datenleitungen können auch zusammengefasst werden

CS = chip select (low active)

DEMUX (1 zu 4) Decodierer

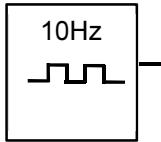


B	A	\overline{CS}	Y3	Y2	Y1	Y0
X	X	1	0	0	0	0
0	0	0	0	0	0	S
0	1	0	0	0	S	0
1	0	0	0	S	0	0
1	1	0	S	0	0	0

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

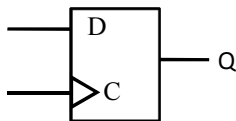
2.3 Schaltwerke

Taktgenerator



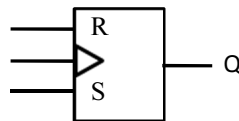
Flip-Flops

D-Flip-Flop



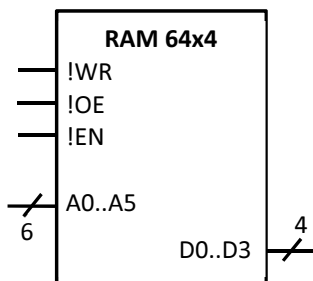
Takt	D	Q ⁿ⁺¹
↑	0	0
↑	1	1
sonst	X	Q ⁿ

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

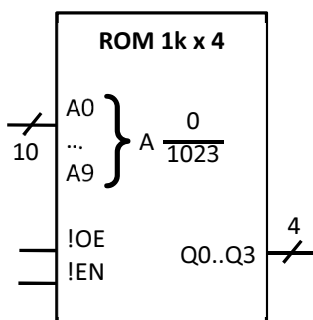
RAM



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**: Hochohmig
- **OE=0**: Speicherinhalt lesen
- **EN=0**: aktiviert den Baustein

ROM

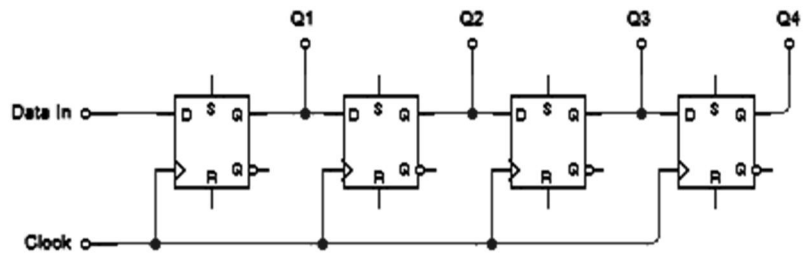
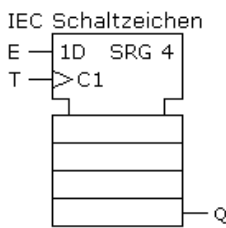


Festwertspeicher mit 1024 (1KiBi) mal 4 Bit

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

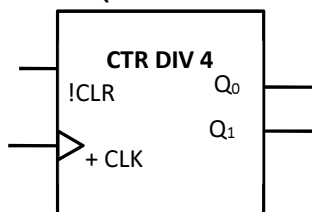
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Schieberegister



Beispiel: Seriell In => Parallel Out

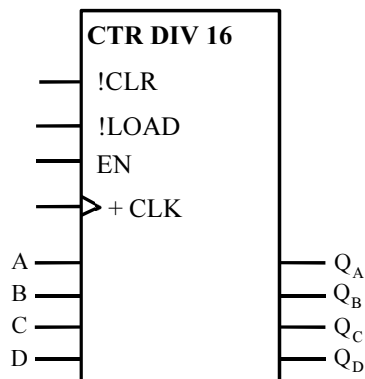
Zähler (Blockschaltbild)



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

- **CTR**: Zähler (counter)
- **DIV 4**: 4 verschiedene binäre Zustände
- **CLR = 0** setzt den Counter auf den Wert 0 zurück
- **Q_n** gibt den Zählerzustand aus

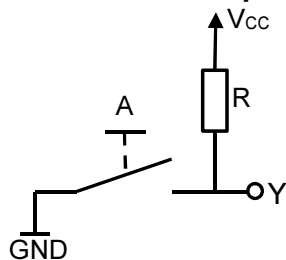
Zähler (4-Bit)



- **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

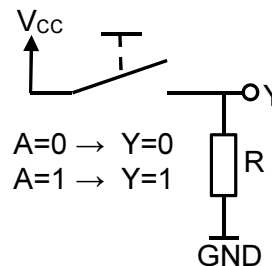
2.4 Sensoren

Taster mit Pull-Up-Widerstand



A=0 → Y=1
A=1 → Y=0

Taster mit Pull-Down Widerstand



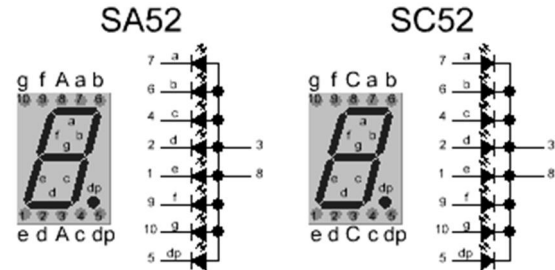
A=0 → Y=0
A=1 → Y=1

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

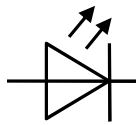
2.5 Aktoren

7-Segmentanzeige

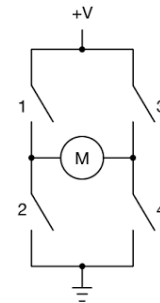
Common Anode ↔ Common Cathode



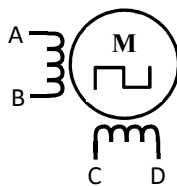
LED



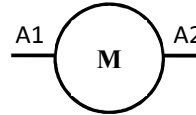
H-Brücke



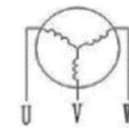
Schrittmotor



DC-Motor

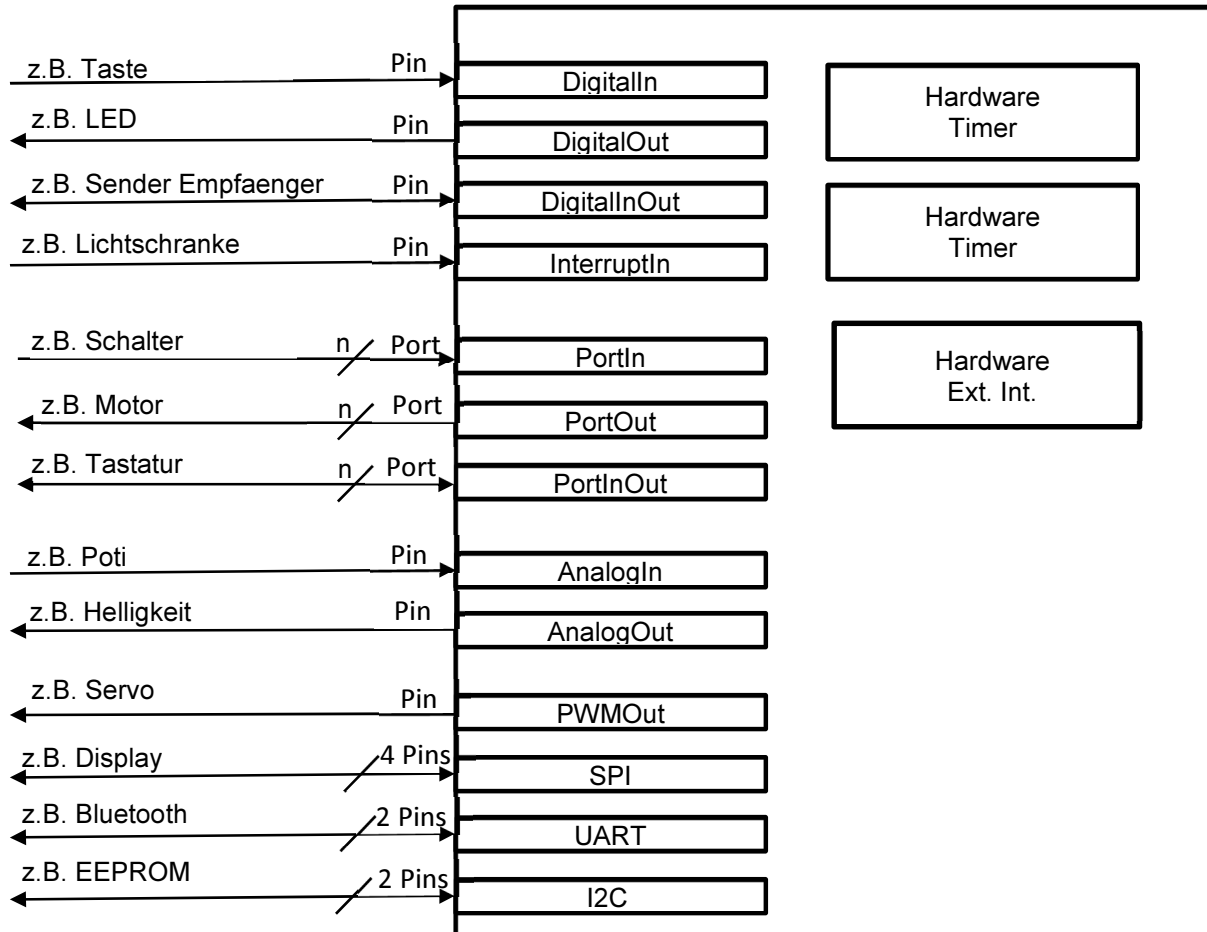


BLDC-Motor



3 Hardware - Mikrocontrollertechnik

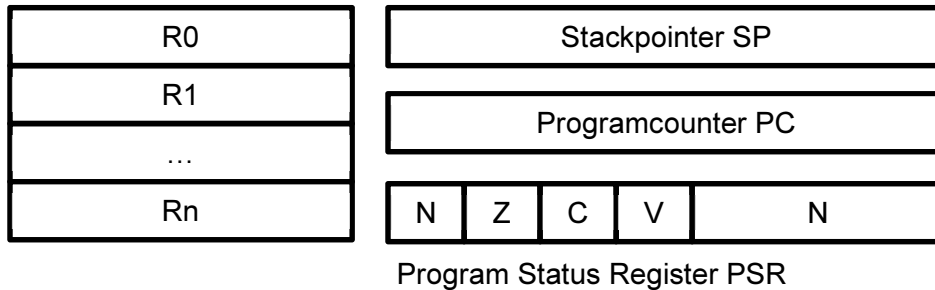
3.1 Blockschaltbild „Prüfungscontroller“



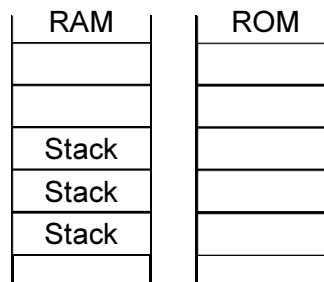
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

3.2 Prozessorarchitektur

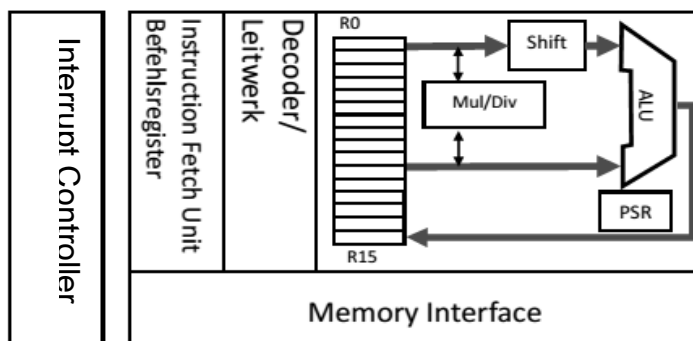
Programmiermodell



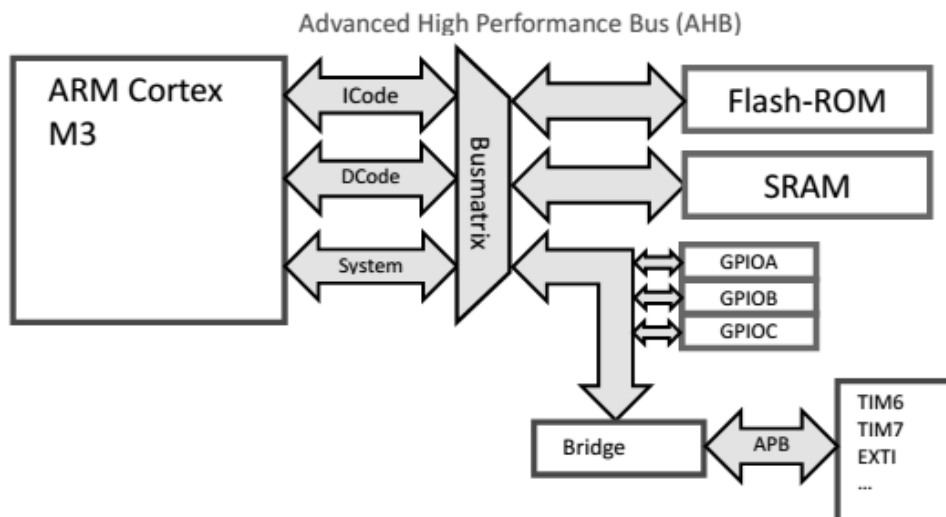
- N** Negative - Flag
- Z** Zero - Flag
- C** Carry - Flag
- V** Overflow - Flag



Blockbild Prozessorkern CPU

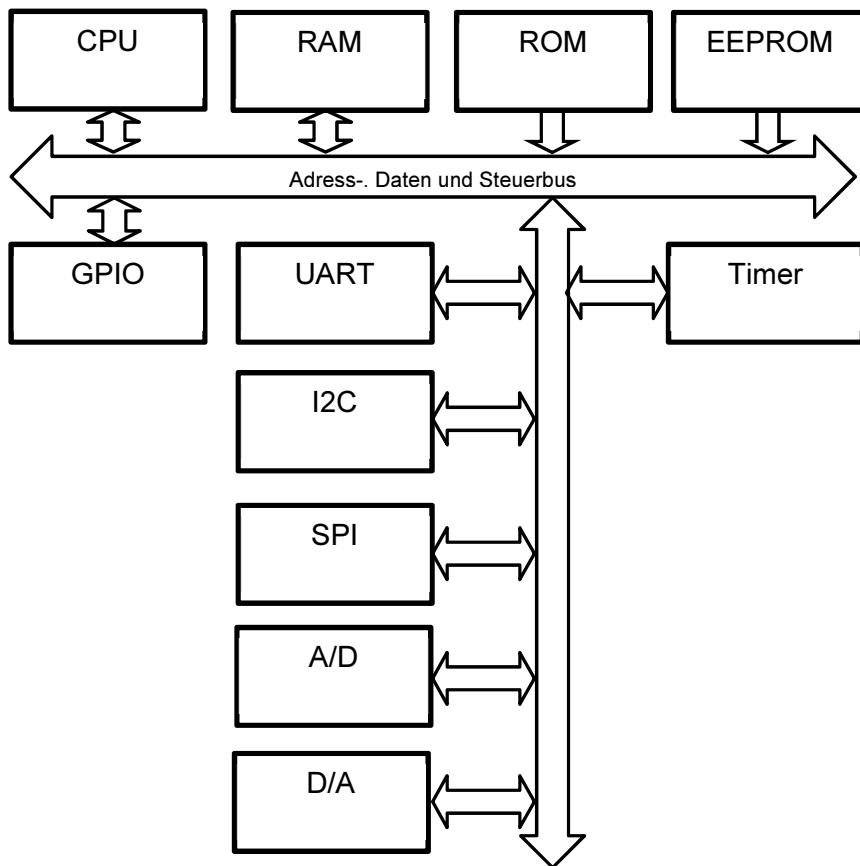


Blockbild Mikrocontroller

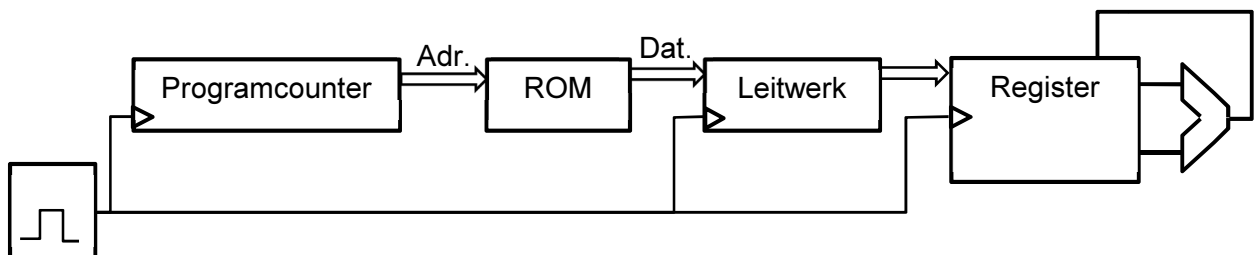


Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Blockschaltbild Mikrocontroller

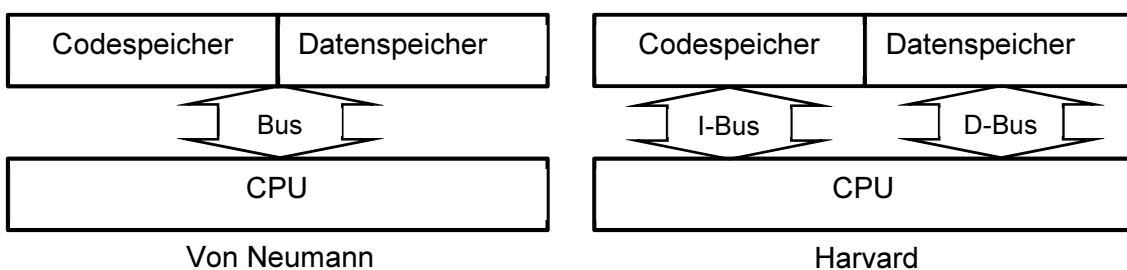


Befehlspipeline einer RISC-CPU



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

Speicherarchitektur



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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
```


Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

4.3 Datentypen

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).

Komplexe Datentypen

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

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

Liste<Typ>
...
+Liste<Typ>() +anzahlElemente():GZ +gib(pIndex:GZ):Typ +ersetzen(pIndex:GZ,pElement:Typ) +einfuegen(pIndex:GZ,pElement:Typ) +anhaengen(pElement:Typ) +verketten(pListe>Liste<Typ>) +entfernen(pIndex:GZ):Typ +entfernenElement(pElement:Typ) +enthaelt(pElement:Typ):Boolean +kopieren():Liste<Typ> ...

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 gibt das gelöschte Objekt vom Datentyp Typ zurück. Die Operation entfernenElement wird mit einem Argument vom Datentyp Typ aufgerufen. Sie sucht das übergebene Objekt in der Liste von vorne und löscht das erste gefundene Objekt, falls vorhanden.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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>() FÜR i←0 BIS 2 SCHRITT 1 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

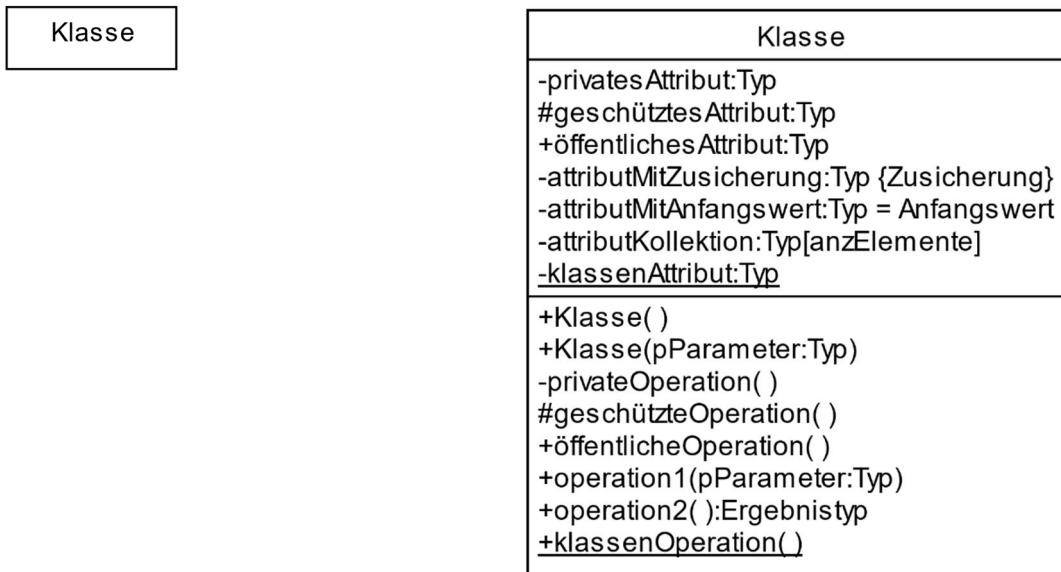
Standardnotation	Bedeutung
highscore ← NEU GZ[10]	Feld für 10 Highscores anlegen.
highscore[0] ← 15	Ersten Highscore auf 15 setzen.

Notationen für fehlende Referenz

Standardnotation	Alternative Notation	Bedeutung
gewinner ← NICHTS	gewinner ← NULL gewinner ← NONE	Objektreferenz, die auf kein Objekt verweist.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

4.4 Klassen



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.

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}.

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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><:Rückgabety>

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.

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
        ABRUCH
    ENDE WENN
ENDE FÜR
WENN gefunden = falsch
    neuePerson ← NEU Person(pName)
    personen.anhaengen(neuePerson)
ENDE WENN
RÜCKGABE gefunden
    
```

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Assoziationen, Rollennamen und Multiplizitäten

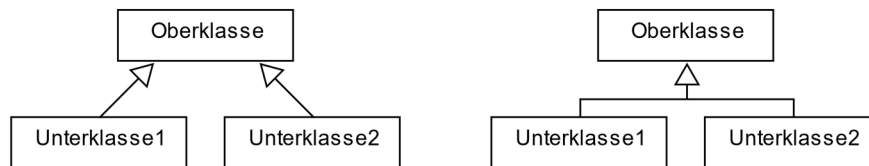


Gerichtete Assoziation

Bidirektionale Assoziation

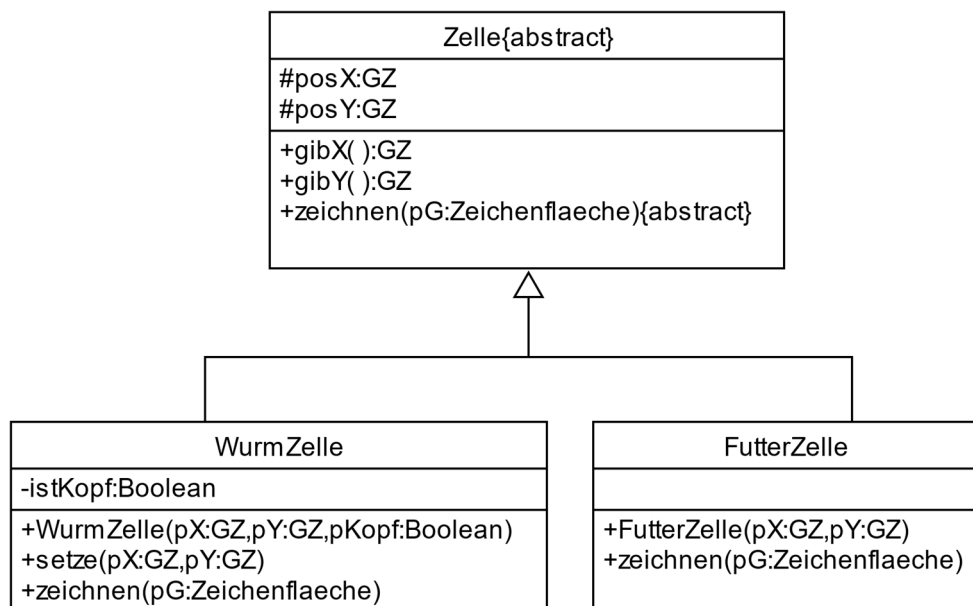
Multiplizität	Bedeutung
1	genau 1
0..1	0 oder 1
3..6	3, 4, 5 oder 6
*	0 bis viele
2..*	2 bis viele

4.5 Vererbung

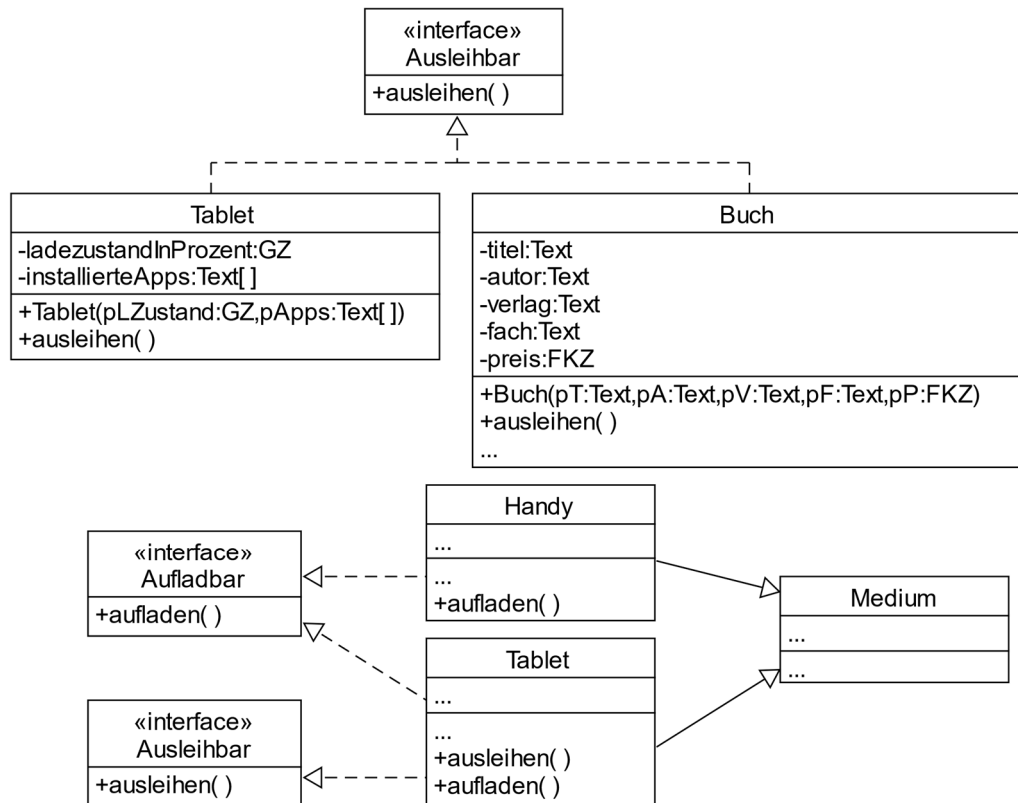


Oberklassen sind Generalisierungen und Unterklassen Spezialisierungen.

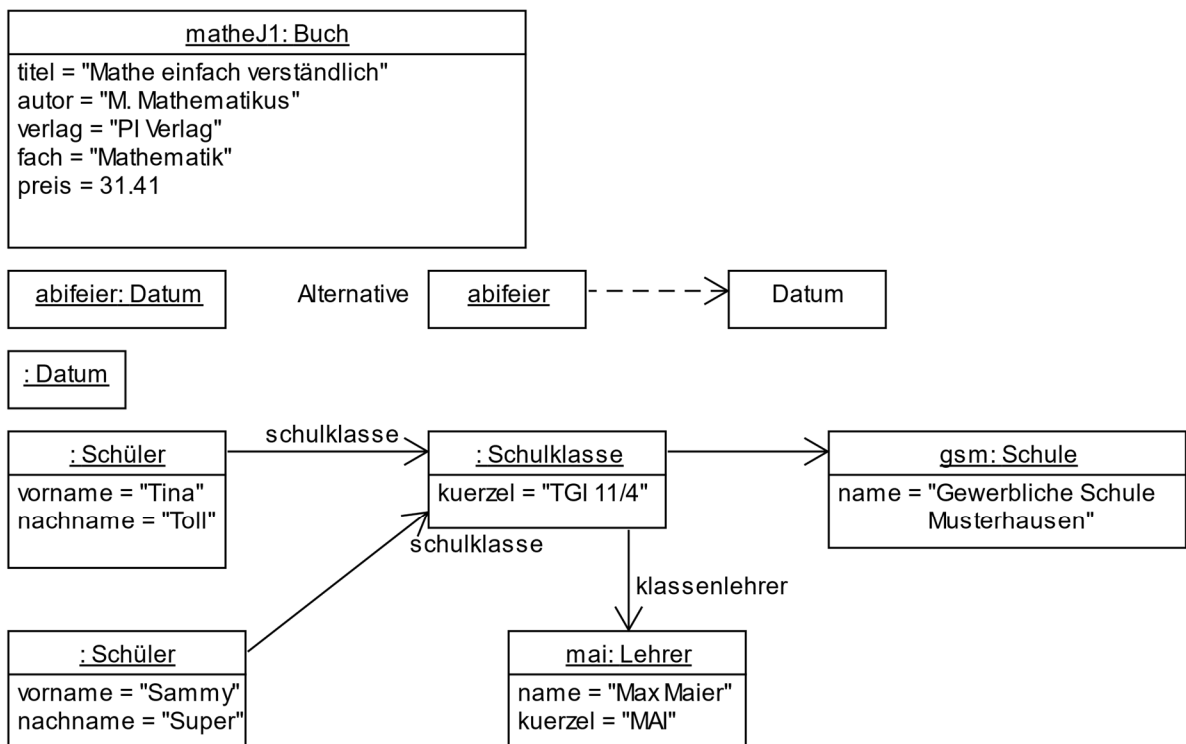
4.6 Abstrakte Klassen und Schnittstellen



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik



4.7 Objektdiagramme



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

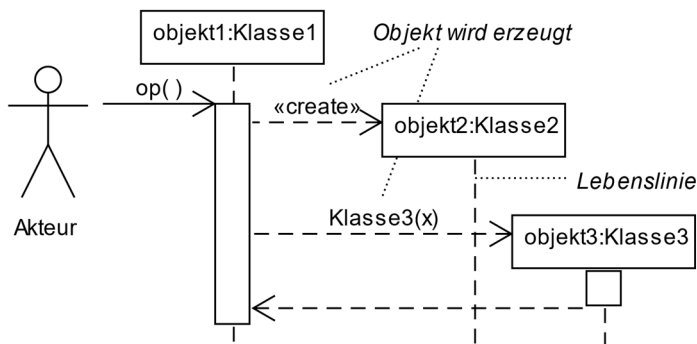
4.8 Sequenzdiagramme

Allgemeines:

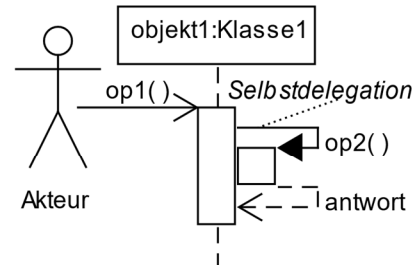
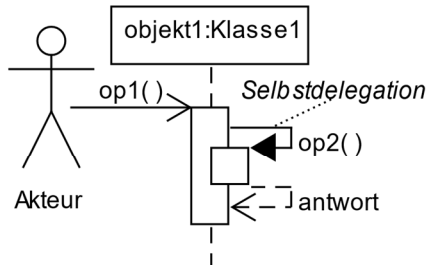
Es wird nicht zwischen unterstrichenen und nicht-unterstrichenen Objekten im Sequenzdiagramm unterschieden.

Erzeugung von Objekten

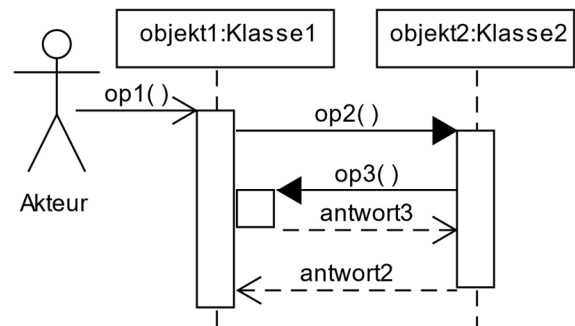
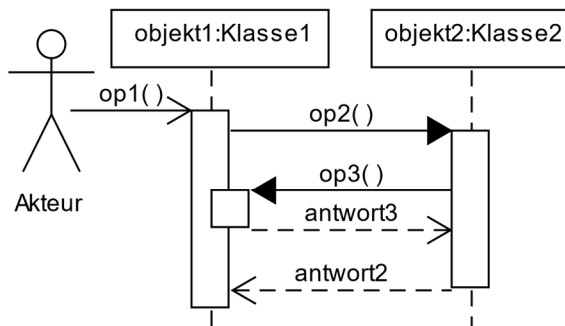
Ein Objekt kann im Sequenzdiagramm immer mit einem spezifischen Konstruktor erzeugt werden. Ist die Auswahl des Konstruktors nicht bedeutsam, so kann die Objekterzeugung durch <<create>> dargestellt werden.



Selbstdelegation (alternative Darstellungen)

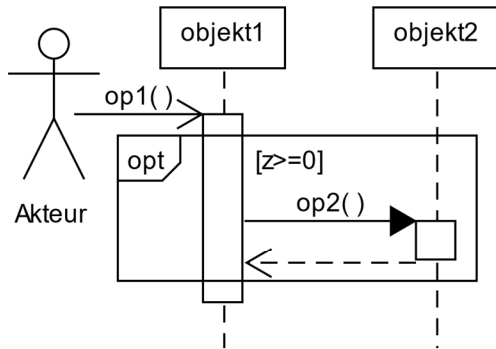


Wechselseitige Botschaften (alternative Darstellungen)

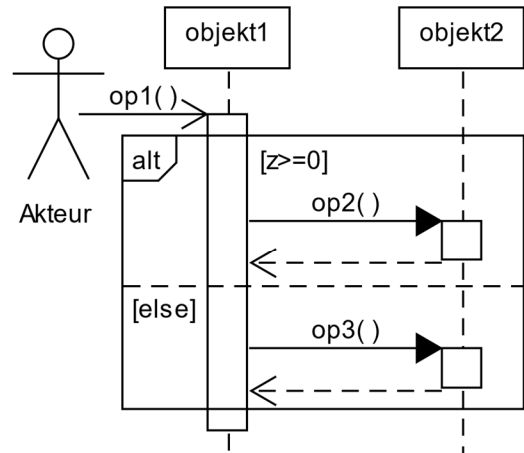


Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

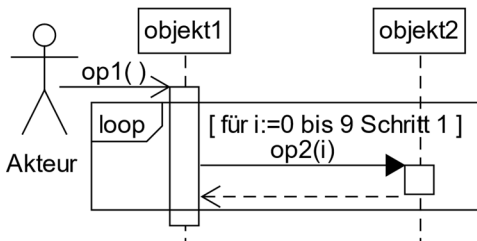
Option – einseitige Verzweigung



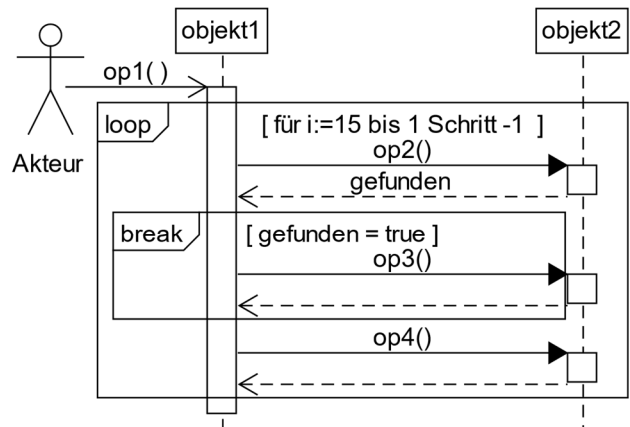
Alternative – mehrseitige Verzweigung



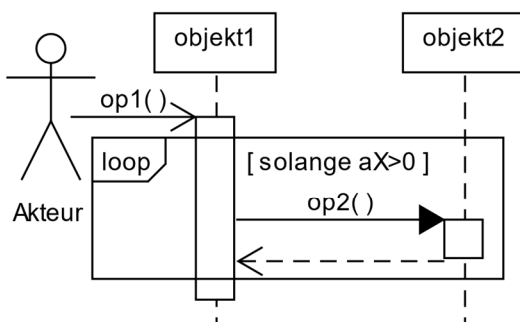
Zählschleife



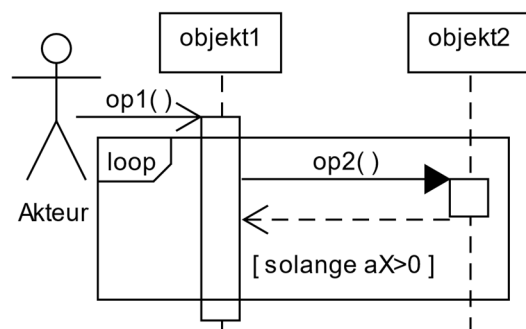
Schleife mit Abbruch



Kopfgesteuerte Schleife

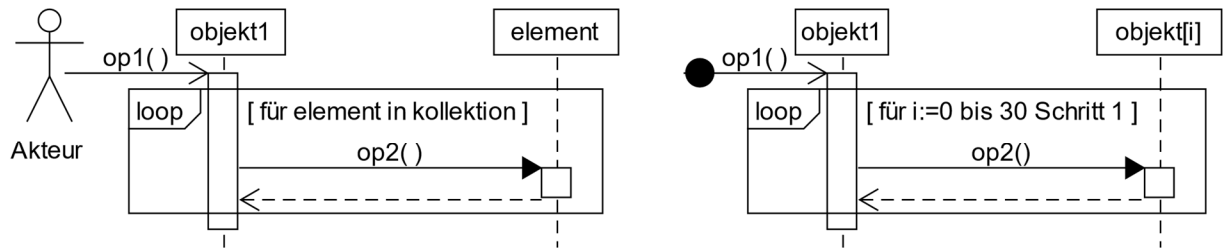


Fußgesteuerte Schleife



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Schleife über Kollektion



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

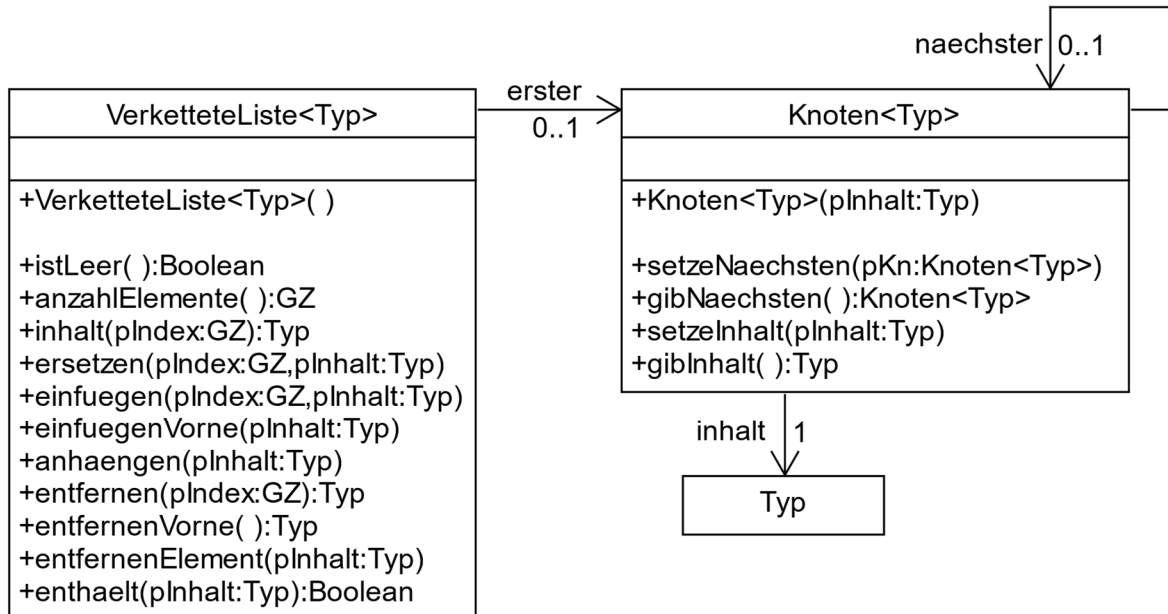
4.9 Zustandsdiagramme

Zustandsdiagramme siehe Kapitel 1

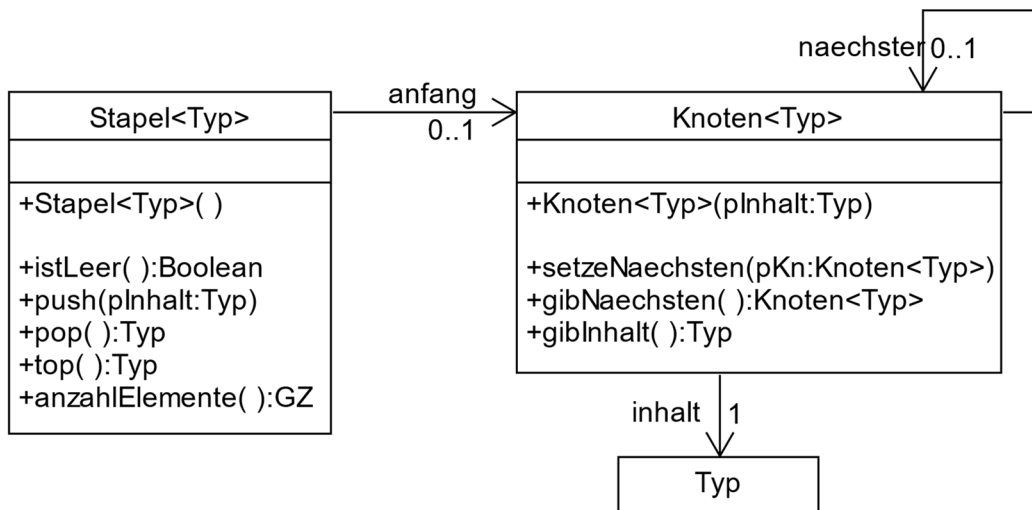
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

5 Datenstrukturen

5.1 Verkettete Liste

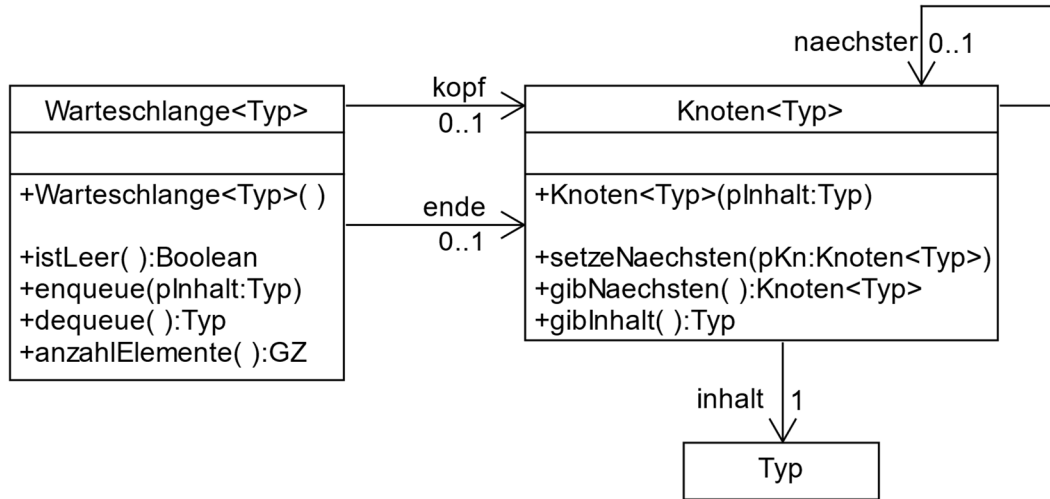


5.2 Stapel



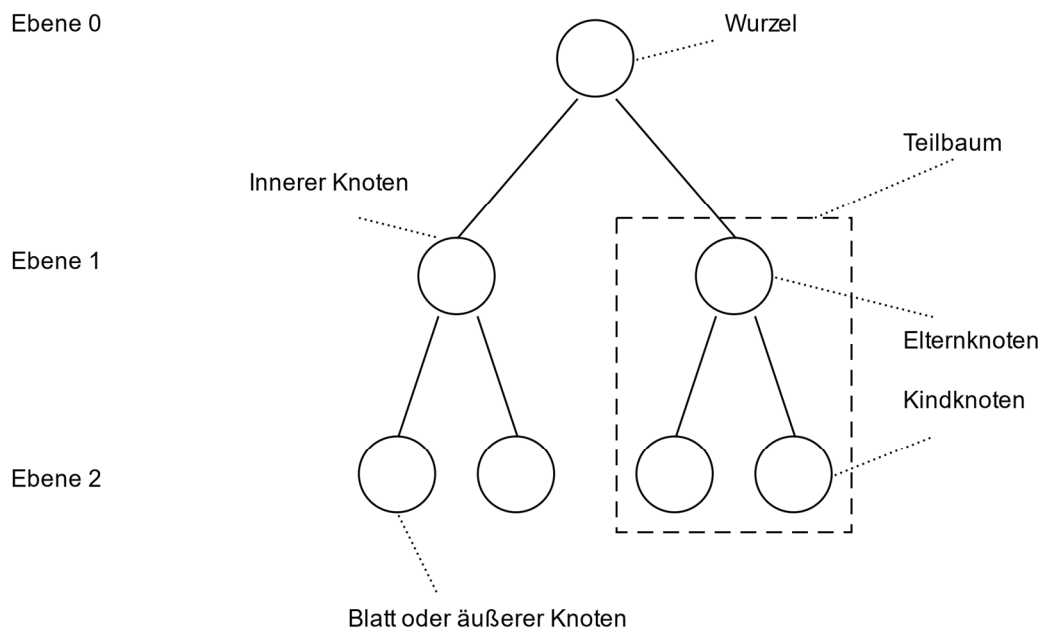
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

5.3 Warteschlange



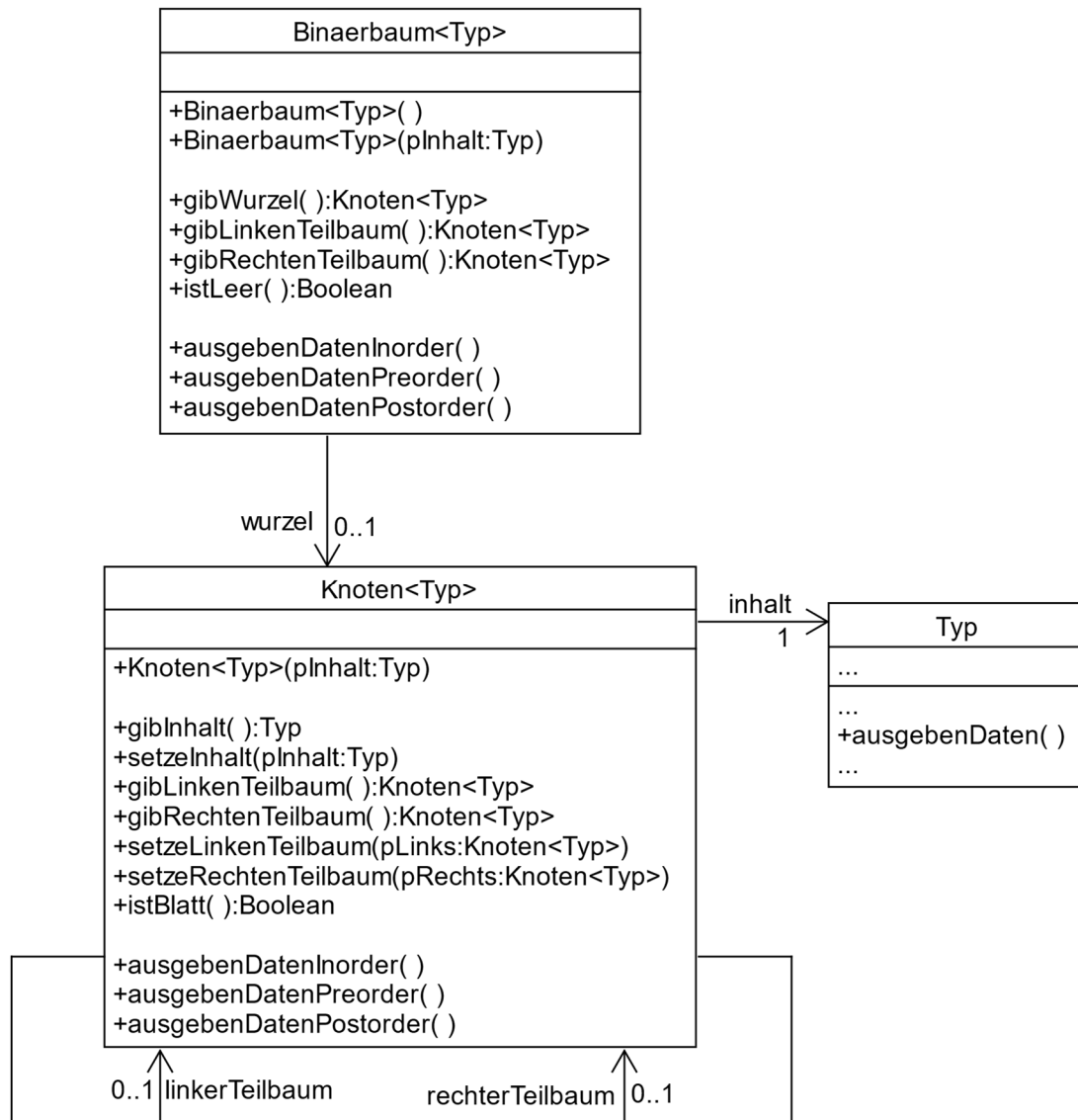
5.4 Binärbaum

Beispiel für einen Binärbaum der Tiefe 3



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Datenstruktur



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 $P(p_1 | \dots | p_n)$ und $Q = (q_1 | \dots | q_n)$

- Euklidische Distanz $d(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$
- Manhattan-Distanz $d(P, Q) = \sum_{i=1}^n |p_i - q_i|$
- Maximum-Distanz $d(P, Q) = \max(|p_i - q_i|)$

6.2 Gini-Unreinheit

Für eine (ausgewählte) Menge von Datensätzen D und einem Ziel-Feature mit k möglichen Ausprägungen ist die **Gini-Unreinheit** (auch: Gini-Koeffizient, Gini-Index, Gini Impurity) wie folgt definiert:

$$Gini(D) = 1 - \sum_{i=1}^k (p_i)^2$$

wobei p_i die relative Häufigkeit der i -ten Ausprägung des Ziel-Merkmals ist.

Mit $Gini(F = v)$ bezeichnen wir die Gini-Unreinheit der Auswahl von Datensätzen, bei denen das Merkmal/Feature F den Wert v hat.

Ein Feature F kann verschiedene Werte $v \in V_f$ annehmen. Tritt ein bestimmter Wert v mit der relativen Häufigkeit p_v auf, dann berechnet sich die **gewichtete Gini-Unreinheit** für das Feature F folgendermaßen:

$$Gini(F) = \sum_{v \in V_f} p_v \cdot Gini(F = v)$$

6.3 Normalisierung von Daten

Normalisierung eines Werts x , d.h. Abbildung in den Wertebereich $[0; 1]$:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

wobei x_{min} der kleinste und x_{max} der größte Wert des entsprechenden Merkmals ist.

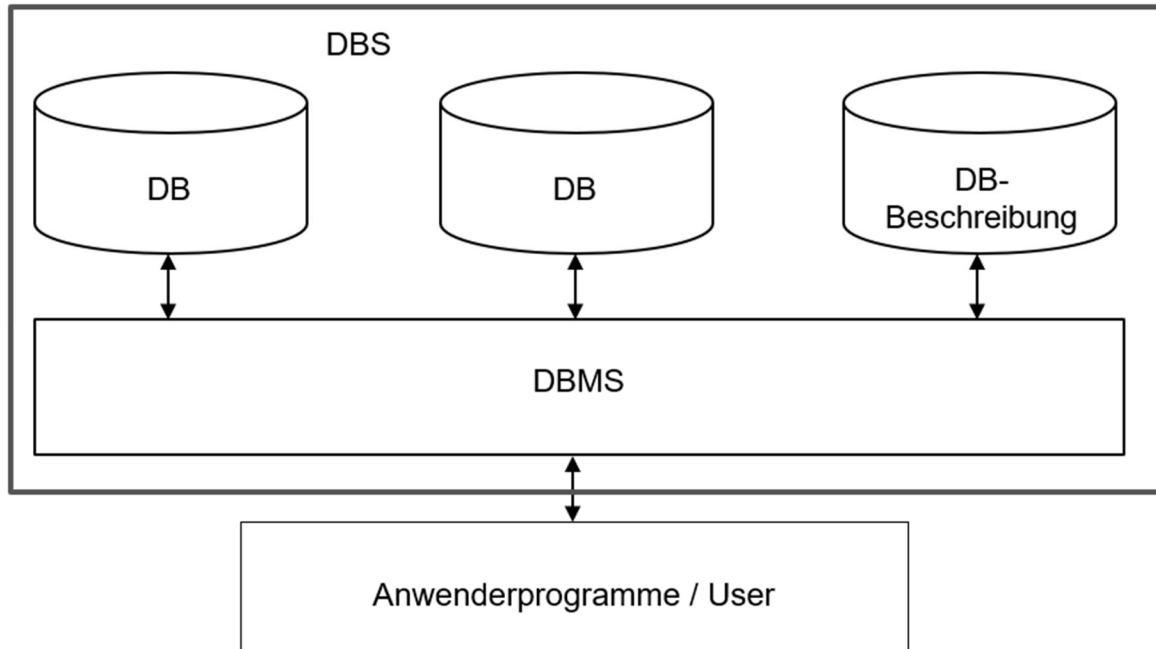
Bsp. Werte: 7; 1; -3; 12; 0; 4 $\rightarrow x_{max} = 12; x_{min} = -3$
 Normalisierte Werte: $\frac{2}{3}; \frac{4}{15}; 0; 1; \frac{1}{5}; \frac{7}{15}$

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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

7 Datenbanken

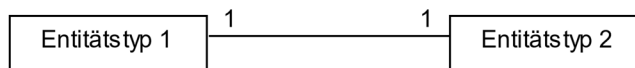
7.1 Datenbankmanagementsystem



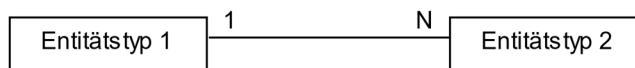
DBS = Datenbanksystem DBMS = Datenbankmanagementsystem DB = Datenbank

7.2 Entity-Relationship-Diagramm (ER-Diagramm)

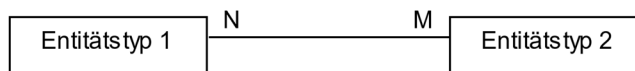
1:1 Beziehung



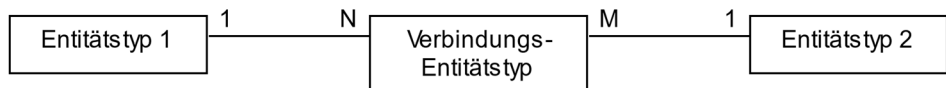
1:N Beziehung



N:M Beziehung



N:M Beziehung aufgelöst



Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

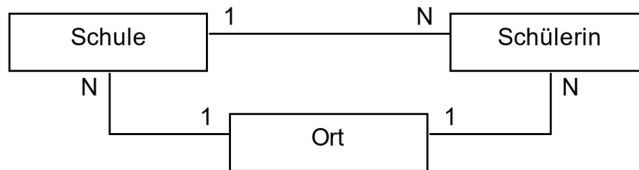
7.3 Relationenmodell

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

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;

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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;

Verbund von Tabellen

Inner Join

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)
 Laborübung(LID, Thema, Dauer)

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

Anmerkung: Tabellennamen können in FROM durch Aliase abgekürzt werden.

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

SELECT Vorname, Name, Datum, Thema, Dauer
 FROM Schüler S
 INNER JOIN Teilnahme T ON S.SID = T.SID
 INNER JOIN Laborübung L ON L.LID = T.LID;

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;

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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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 INNER JOIN Teilnahme T ON S.SID = T.SID
WHERE 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 INNER JOIN Teilnahme T ON S.SID = T.SID
WHERE 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 INNER JOIN Teilnahme T ON S.SID = T.SID;

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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)

Punktendurchschnitte pro Klasse
 SELECT Klasse, AVG(Punkte) AS "Gesamtpunktzahl pro Klasse"
 FROM Schüler S INNER JOIN Teilnahme T ON S.SID = T.SID
 GROUP BY Klasse;

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

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)

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

Komplette SQL-Anweisung

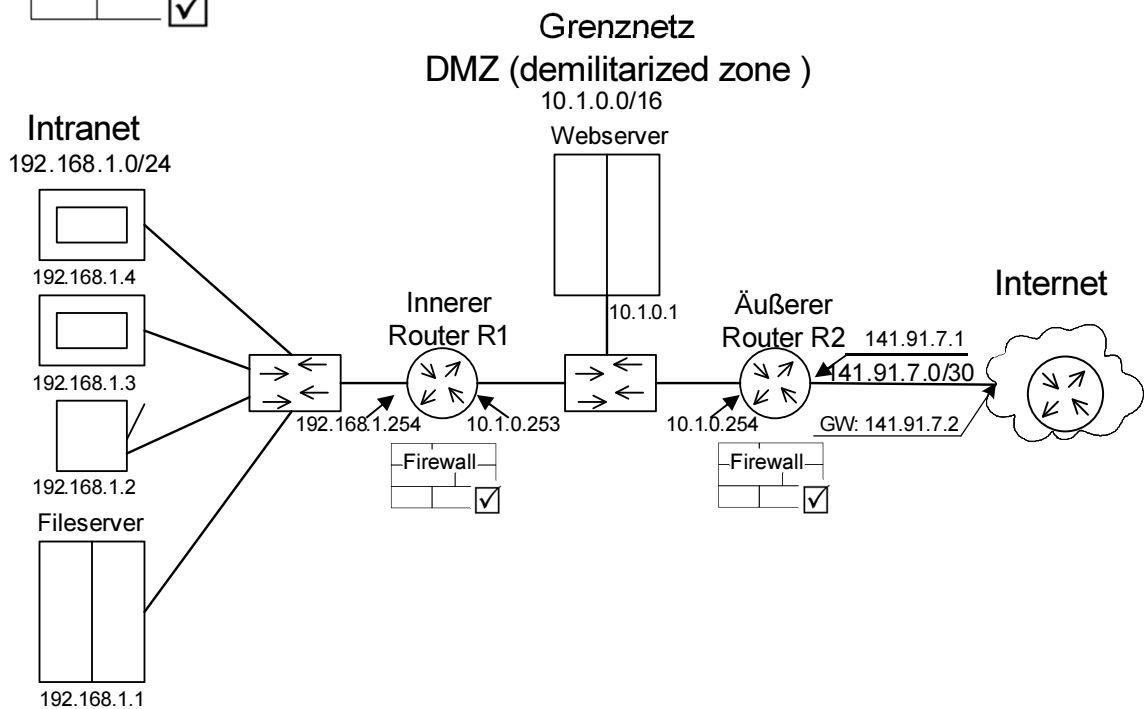
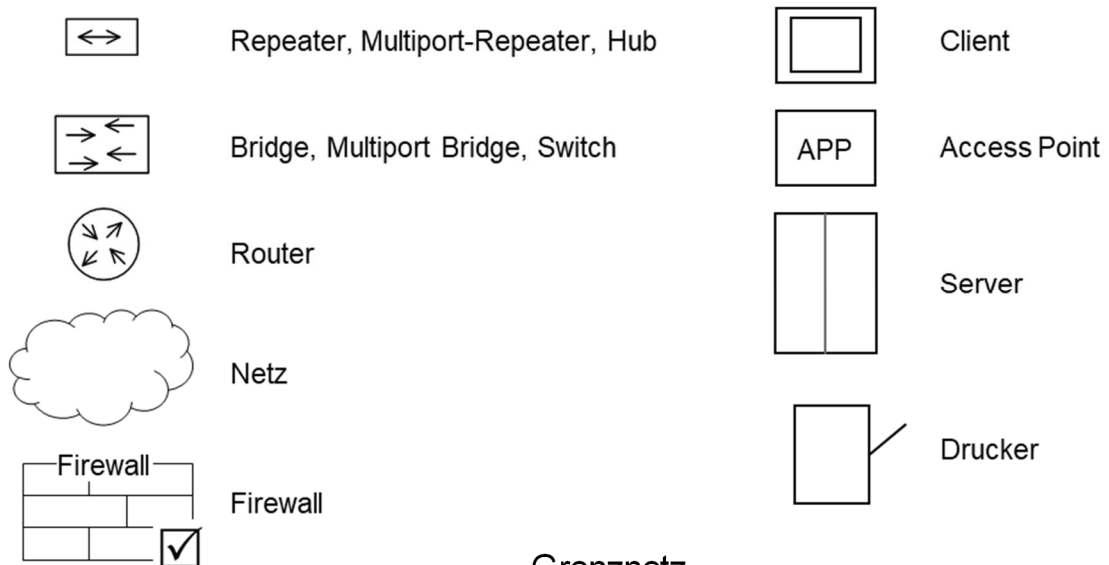
Syntax: SELECT ...
 FROM ...
 WHERE ...
 GROUP BY ...
 HAVING ...
 ORDER BY ... ;

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

8 Vernetzte Systeme

8.1 Netzwerktechnik

Netzwerksymbole



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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Aufbau IPv4-Adresse

IP-Adresse (dotted-decimal-format): z.B. 177 . 17 . 223 . 1
 IP-Adresse (binär): 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

Aufbau IPv6-Adresse

IP-Adresse (hexadezimal): z.B. 2001:07c0:8280:0253:0000:0000:0000:0020

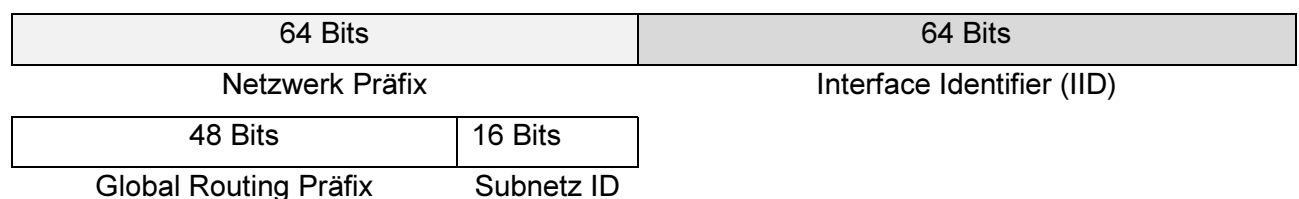
16 Bit

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

Adressformat:



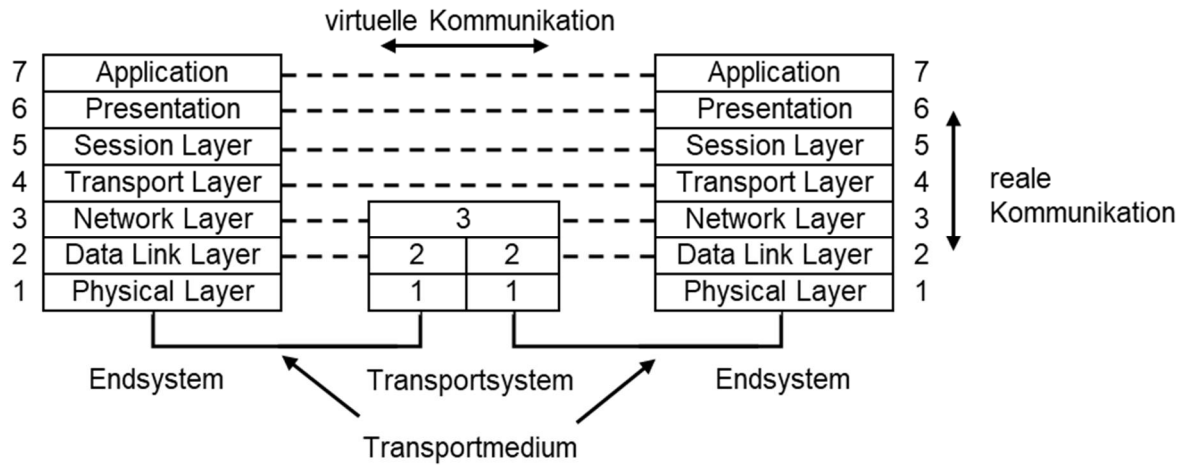
Netzwerk-Präfix: 2001:07c0:8280:0253 → Global Routing Präfix 2001:07c0:8280: 0253 → Subnetz Identifier	Interface Identifier: xxxx:xxxx:xxxx:xxxx: 0000:0000:0000:0020
--	--

Adressbereich-Zuweisung:
 2001:07c0:8280:0253::**64**
 2001:07c0:8280:0200::**56**
 2001:07c0:8280::**48**
 2001:07c0::**32**

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

8.2 Schichtenmodelle

ISO-OSI-7-Schichtenmodell



TCP-IP-Schichtenmodell

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

8.3 Header

Ethernet II

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

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
20 ...	Optionen (mit evtl. Füllzeichen)

IPv4-Paketstruktur:

IPv4-Header 20-60 Bytes	Upper Layer Protocol Data Unit (TCP, UDP, ICMP, ...)
----------------------------	---

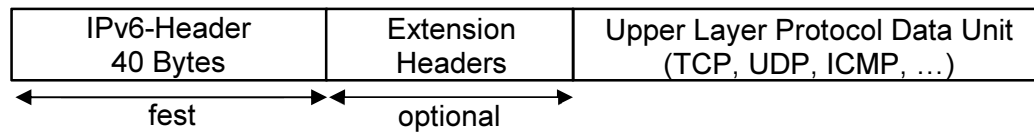
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

IPv6-Header

← variabel →

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:



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	Fenstergröße
16-17	Prüfsumme
18-19	Urgent Zeiger
20 ...	Optionen (evtl. mit Füllzeichen)

UDP –Header

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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Firewall-Regelkatalog (vereinfacht)

Firewalls verfügen typischerweise über die folgenden Grundfunktionen: sie lassen Pakete passieren, sie verwerfen Pakete und sie verweigern den Transport. Diese Funktionen werden auf eine Reihe von Paket-Attribute angewandt. Dabei wird sowohl der eingehende als auch der ausgehende Datenverkehr gefiltert. Diese grundsätzliche Funktionsweise wird im Folgenden abstrahiert an einigen Beispielen gezeigt.

Befehl	Beschreibung
ALLOW	Pakete, die dem angegebenen Kriterium genügen, dürfen die Firewall passieren.
DROP	Eingehende Pakete werden verworfen, der Versender erfährt davon nichts.
REJECT	Eingehende Pakete werden abgelehnt, der Absender bekommt eine entsprechende Antwort.

Die obigen Funktionen können auf die folgenden Paketeigenschaften angewandt werden (Auswahl und vereinfacht):

Parameter	Beschreibung
Interface	WAN, LAN, VLANxx (z.B. VLAN01, VLAN10)
Version	Protokollversion (IPv4 oder IPv6 oder beides IPv4+6)
Protokoll	Protokolltyp (z.B. UDP, TCP)
Quelle	IP-Adresse, Adressbereich, lokale Rechnernamen, Domainnamen, MAC-Adressen. Invertierung mit vorangestelltem Ausrufezeichen (!) möglich.
Quellport(s)	Wert(e) des Quellports im Paket (einzelne Ports, z.B. 80, mehrere Ports durch Komma getrennt, z.B. 80, 8080 oder Portbereiche, z.B. 9000-65535)
Ziel	wie Quelle, nur für die Ziele im Paket
Zielport(s)	wie Quellport, nur für den Zielport

Für Quelle, Ziel, Quellport und Zielport kann vereinfacht ANY (Alle erlaubt) eingestellt werden. Die Syntax einer Regel lautet:

Befehl Interface Version Protokoll Quelle Quellport Ziel Zielport

Beispiele:

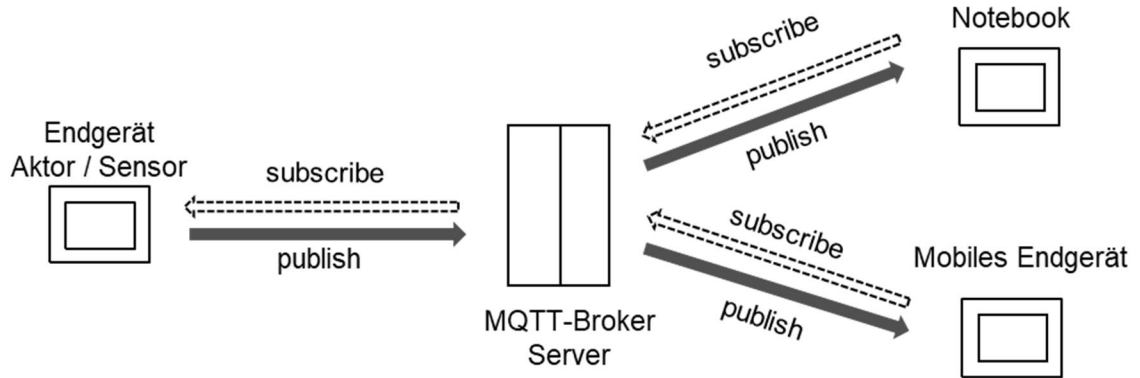
ALLOW VLAN10 IPv4 ANY ANY ANY ANY ANY	Aus dem (internen) VLAN10 heraus wird jeglicher Datenverkehr ohne Einschränkung zugelassen.
DROP LAN IPv4+6 dbserver TCP ANY ANY ANY	Alle TCP-Pakete des Rechners mit dem Namen dbserver im internen Netz werden verworfen
DROP LAN IPv4+6 172.16.0.0/24 ANY 80, 8080 ANY ANY	Alle Pakete vom Port 80 aus dem Netzbereich 172.16.0.0/24 werden verworfen.

Zur Vereinfachung kann davon ausgegangen werden, dass Antworten auf eine erlaubte Anfrage in das geschützte Netz oder aus dem geschützten Netz heraus immer durchgelassen werden (Stateful Packet Inspection).

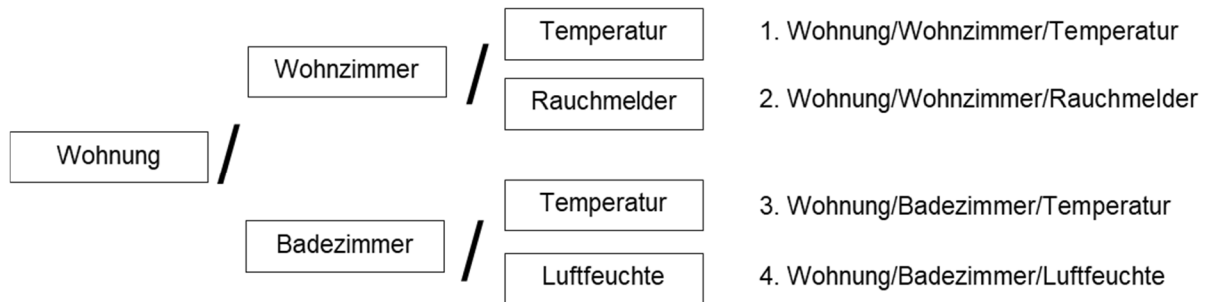
Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

8.4 Internet der Dinge (IoT)

MQTT-Protokoll (Message Queuing Telemetry Transport)



Topicstruktur:



Multi-Level-Wildcard:

Wohnung/Wohnzimmer/#

Single-Level-Wildcard: +

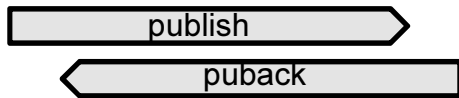
Wohnung+/Temperatur

Qualitätsstandards:

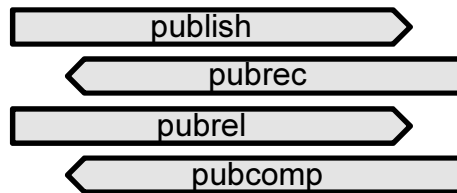
QoS 0



QoS 1



QoS 2



Ports:

1883 : MQTT, unverschlüsselt

8883 : MQTT, verschlüsselt

8884 : MQTT, verschlüsselt, Client Zertifikat notwendig

8080 : MQTT über WebSockets, unverschlüsselt

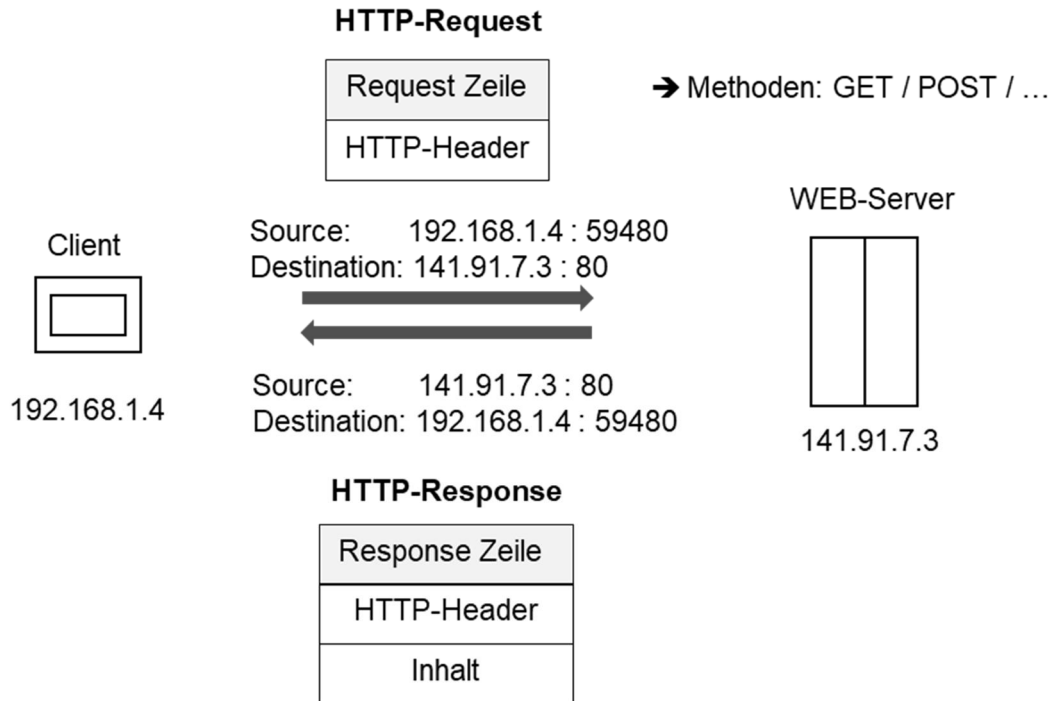
MQTT –Header: (Beispiel - Publish Message)

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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

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

Abiturprüfung ab 2024	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Request HTTP 1/1

Methode	Pfad	Protokoll
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"> • Accept-Charset: → Welche Zeichensätze der Client anzeigen kann. • Accept-Encoding: → Welche komprimierten Formate der Client unterstützt. • Accept-Language: → Gewünschte Sprachversion
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

Protokoll	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 z.B. 200 – OK
300 ... 399:	Client-Anfrage umgeleitet z.B. 301 – Moved Permanently 302 – Moved Temporarily
400 ... 499:	Fehlen des Dokuments z.B. 403 – Forbidden 404 – Not Found
500 ... 599:	Serverfehler