# Aufgabe 1: 3-Schichten-Architektur

Benennen Sie die drei Schichten einer 3-Schichten-Architektur und beschreiben Sie kurz deren jeweilige Funktion.

Schicht 1:

Schicht 2:

Schicht 3:

Name:_____ Matr.-Nr: _____

# Aufgabe 2: Grammatiken

Gegeben sei eine Grammatik wie folgt ($\varepsilon$ ist Leersymbol):

G = <Startsymbol, Terminale, Nicht-Terminale, Regeln> mit

Startsymbol = $Start$

Terminale = $\{a, b, c, d\}$

Nicht-Terminale = $\{X, Y\}$

Regeln = $\{ Start \rightarrow XY,$
$X \rightarrow aX,$
$X \rightarrow bX,$
$X \rightarrow b,$
$Y \rightarrow cY$
$Y \rightarrow d \}$

Entscheiden Sie, ob folgende Wörter in der durch die Grammatik definierten Sprache enthalten sind:

- $\varepsilon$ ja / nein
- aabccb ja / nein
- abcd ja / nein
- aabbaacb ja / nein
- acdc ja / nein
- aabbabcd ja / nein
- abc ja / nein
- aabccd ja / nein
- bd ja / nein
- aabbbc ja / nein

Bestimmen Sie die Wörter, die durch die Grammatik akzeptiert werden in der Art $a^n b^n$:

Name:_____ Matr.-Nr: _____

# Aufgabe 3: Endliche Automaten

Gegeben sei eine Doppeltürschleuse, wo die zweite Tür sich erst öffnen lässt, wenn sich eine Person durch die erste Tür in die Schleuse begeben hat und die erste Tür durch Schließen des Gatters hinter sich geschlossen ist. Erst wenn die die Person die Schleuse verlassen hat und zweite Tür wieder geschlossen ist, ist die Schleuse wieder im Startzustand. Ein Sensor überprüft, ob sich eine Person in der Schleuse befindet. Bevor ein Gatter geschlossen ist, steht die jeweilige Tür offen. Als Beispiel sei gegeben:

KABA BEYOND SECURITY

Name:_____ Matr.-Nr: _____

# Aufgabe 4: Prolog Unifikation

Gegeben seien jeweils 2 Prolog-Ausdrücke. Sie sollen entscheiden, ob diese beiden Ausdrücke unifizierbar sind und, falls dieses der Fall ist, den durch Substitution erzeugten Ausdruck angeben. Sollte die Unifikation misslingen, geben Sie den Grund an. Als Beispiele sind gegeben (Konstanten fangen mit kleinem, Variablen mit großem Buchstaben an):

- test (X, foo, Y) = test (M, N, M) ☺ mit test (X, foo, X).
- test (kai, X) = test1(kai, udo) ☹ weil test ≠ test1.

Hier nun die Ausdrücke:

- comp(Mainboard, intel(i7)) = comp(asus(p6t), CPU).

- comp(Mainboard, intel(i7)) = comp(asus(p6t), amd(Type)).

- append(X, [4, 5, 7], Y) = append([1,2,3], Z, [a, b, c]).

- append([1, 2, 3], [4, 5, 6], X), X = [a, b, c].

Name:_____ Matr.-Nr: _____

# Aufgabe 5: Prolog Programmanalyse

Gegeben sei ein Prolog-Programm `foo(List1, List2)` wie folgt (`reverse/2` dreht die Reihenfolge einer Liste um):

```
foo(Arg1, Arg2) :-
    reverse(Arg1, Temp1),
    append(Arg1, Temp1, Temp2),
    append(Temp2, Arg1, Arg2).
```

Beschreiben Sie kurz, was das Programm macht und geben Sie jeweils ein Beispiel mit belegtem 1. Argument und variablem 2. Argument wie belegtem 2. Argument und variablem 1. Argument:

Beispiele:

**Bonus-Aufgabe: was macht folgende Funktion:**

```
cool(X,Y) :-
    cool(X, [], Y, Y).

cool([], List, List, []).
cool([First|Rest], List1, List2, [_|Rest2]) :-
    cool(Rest, [First|List1], List2, Rest2).
```

Name:_____ Matr.-Nr: _____