

# Gedächtnisprotokoll – Test01A 16.01.26

---

## Aufgabe 1 – Multiple Choice

\_ / 3 P

Zu jeder Frage gibt es genau **eine richtige Antwort**.

Es gibt **keinen Punktabzug** für falsche Antworten.

Werden **mehr als eine Antwort** angekreuzt, wird die Frage mit **0 Punkten** bewertet.

### a) Welche Testarten sind im V-Modell vorgesehen?

\_ / 0.75 P

- ☐ Komponenten-, System- und Abnahmetests
  - ☐ Integrations-, Last- und Stresstests
  - ☐ Unit-Tests, Modul- und Performanztests
  - ☐ Komponenten- und Systemtests
- 

### b) Was ist das Hauptziel von Refactoring?

\_ / 0.75 P

- ☐ Sicherzustellen, dass sich beobachtetes Verhalten nicht verändert
  - ☐ Neue Funktionen möglichst früh zu validieren
  - ☐ Fehlerursachen im Code eindeutig zu lokalisieren
  - ☐ Die Testabdeckung signifikant zu erhöhen
- 

### c) Für welche Art von Projekten ist das Wasserfallmodell besonders geeignet?

\_ / 0.75 P

- ☐ Große Projekte mit stabilen Anforderungen
  - ☐ Kleine Projekte mit häufig wechselnden Anforderungen
  - ☐ Agile Teams mit inkrementeller Entwicklung
  - ☐ Forschungsprojekte mit unklaren Zielsetzungen
- 

### d) Welche Arten von Metriken werden in der Softwaretechnik unterschieden?

\_ / 0.75 P

- [ ] Produktmetriken und Prozessmetriken
- [ ] funktionale und nicht-funktionale Metriken
- [ ] deterministische und nicht-deterministische Metriken
- [ ] Statistische und Qualitative Metriken

## Aufgabe 2 – Testen und Metriken

**\_/7 P**

### a) Halstead-Metriken

**\_/2 P**

Gegeben ist folgendes Programmfragment:

```
double cube(double x) {  
    return x * x * x;  
}
```

**Berechnen Sie die folgenden Halstead-Metriken:**

Vokabular  $n =$

**\_/0.5 P**

Länge  $N =$

**\_/0.5 P**

Volumen  $V =$

**\_/0.5 P**

Schwierigkeit  $D =$

**\_/0.5 P**

### b) Kontrollflussgraph (CFG)

**\_/5 P**

Gegeben ist das folgende Java-Programm:

```
01 int klausurCfg(int a, int b) {  
02     int x = a + b;  
03     if (a > 0 &&  
04         b > 0) {  
05         while (x < 10) {  
06             x = x + 1;  
07             x = x + 2;  
08         }  
09     } else {  
10         return -1;  
}
```

```
11     }
12     if (x % 2 == 0) {
13         x = x + 5;
14     } else {
15         x = x - 3;
16     }
17     return x;
18 }
```

### Aufgabe:

Zeichnen Sie den vollständigen Kontrollflussgraphen (CFG) des Programms. Beschriften Sie die Knoten mit der zugehörigen Zeilennummer. Sie dürfen Knoten zusammenfassen, wenn aus dem Namen des Knoten eindeutig hervorgeht, welche Zeilen er zusammenfasst. Zeilen, die ausschließlich geschweifte Klammern oder ein `else` enthalten, werden nicht als eigene Knoten gezählt.

## Aufgabe 3 – Prolog

    /10 P

### Gegebene Datenbank

```
% autor(Person, Buch).
autor(b1, 'Cornelia Funke').
autor(b2, 'Cornelia Funke').
autor(b3, 'Cornelia Funke').
autor(b4, 'Suzanne Collins').
autor(b5, 'Suzanne Collins').

% titel(Buch, Titel).
titel(b1, 'Tintenherz').
titel(b2, 'Tintenblut').
titel(b3, 'Tintentod').
titel(b4, 'Die Tribute von Panem').
titel(b5, 'Catching Fire').

% seiten(Buch, Seitenzahl).
seiten(b1, 600).
seiten(b2, 640).
seiten(b3, 720).
seiten(b4, 384).
seiten(b5, 391).
```

```
% bewertung(Buch, Wert).  
bewertung(b1, 8).  
bewertung(b2, 9).  
bewertung(b3, 7).  
bewertung(b4, 9).  
bewertung(b5, 8).  
  
% reihe(Reihentitel, Buchliste).  
reihe('Tintenherz', [b1, b2, b3]).  
reihe('Panem', [b4, b5]).
```

**a) Formulieren Sie eine Query, die das Buch mit dem Titel 'Tintenherz' ausgibt.**

**\_ / 0.5 P**

erwartete Ausgabe:

X = b1

?-

**b) Formulieren Sie eine Query, die die Bewertung des Buches mit dem Titel 'Tintenherz' liefert.**

**\_ / 0.5 P**

erwartete Ausgabe:

X = 8

?-

**c) Formulieren Sie eine Query, die alle Bücher von Cornelia Funke als Liste von Titeln zurückgibt.**

**\_ / 1 P**

erwartete Ausgabe:

Liste = ['Tintenherz', 'Tintenblut', 'Tintentod']

?-

**d) Definieren Sie ein Prädikat `kurzesCollinsBuch(B)` ,** **\_/1 P**  
**das alle Bücher von Suzanne Collins liefert, die weniger als 500**  
**Seiten haben.**

?-

**e) Definieren Sie ein Prädikat `istInReihe(B, R)` ,** **\_/1.5 P**  
**das für ein Buch B den Namen der Reihe R liefert, in der das Buch**  
**enthalten ist.**

Beispiel-Query:

```
?- istInReihe(b2, R).
```

erwartete Ausgabe:

```
R = 'Tintenherz'
```

?-

**f) Definieren Sie ein Prädikat `gleicherAutor(L, P)` ,** **\_/2 P**  
**das genau dann wahr ist, wenn alle Bücher in der Liste L von der**  
**Person P geschrieben wurden.**

Beispiel-Query:

```
?- gleicherAutor([b1,b2,b3], 'Cornelia Funke').
```

erwartete Ausgabe:

```
true
```

?-

**g) Definieren Sie ein Prädikat**

**\_/3.5 P**

**durchschnittBewertung(B, D) , wobei B eine Liste von Büchern ist und D deren durchschnittliche Bewertung als Kommazahl darstellt.**

Beispiel-Query:

```
?- durchschnittBewertung([b1,b3,b4], D) .
```

erwartete Ausgabe:

```
D = 8.3333333
```

```
?-
```