



# Klausur „Programmiersprachen 1“

## Schriftlicher Teil

**Dauer:** 90 Minuten, eine vorzeitige Abgabe ist aufgrund des folgenden praktischen Teils am Computer nicht möglich!

**Erlaubte Hilfsmittel:** Taschenrechner

Notieren Sie Ihre Lösungen ausschließlich in dem dafür vorgesehenen Bereich auf dem Papier. Sollten sie **mehr als den vorgesehenen Platz** benötigen verweisen sie von dort auf die **Rückseite**. Auf Rückseiten angefertigte Lösungen müssen eindeutig mit der Nummer der Aufgabe bezeichnet werden.

Liegen **mehrere Lösungswege** zu einer Aufgabe vor, so wird keiner davon bewertet, es sei denn, eine Lösung ist eindeutig als Entwurf bzw. Kladde erkennbar.

Es sind in diesem Teil insgesamt maximal **35 Punkte** zu erreichen. Bei jeder Aufgabe ist die dort **mögliche Punktzahl** angegeben, die somit auch eine ungefähre Aufwandsabschätzung ermöglicht. Sollte eine **Lösung deutlich umfangreicher** sein, als die vorhandene Maximalpunktzahl annehmen lässt, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Ihnen **Aufgabenstellungen missverständlich** erscheinen, so legen Sie bitte Ihre Interpretation der Aufgabe kurz schriftlich dar und lösen Sie die Aufgabe entsprechend.



## 1. Syntaxdiagramm

5 Punkte

Eine Bankverbindung nach dem „International Bank Account Number“-Standard IBAN setzt sich in Deutschland aus vier Teilen zusammen. Bis auf den Ländercode handelt es sich bei allen Bestandteilen des Codes um Ziffern. Diese Teile werden in der folgenden Reihenfolge zusammengesetzt:

1. Der Ländercode lautet stets „DE“.
2. Die zweistellige Prüfsumme
3. Die achtstellige Bankleitzahl.
4. Die zehnstellige Kontonummer.

DE44500105175407324931 wäre nach diesen Regeln eine gültige IBAN.

---

Überführen Sie obige Beschreibung in ein Syntaxdiagramm. Verwenden Sie dabei für jede unterstrichene Bezeichnung eine eigene Regel. Die Regeln Ziffer, Vier Ziffern und Buchstabe können Sie als gegeben voraussetzen und gegebenenfalls verwenden.



## 2. Backus-Naur-Form

Entscheiden Sie für die folgenden Texte jeweils individuell ob sie zu der gegebenen in Backus-Naur-Form notierten Grammatik passen. Zeichnen sie dafür für jeden Ausdruck den sich ergebenden Baum mit der Regel „Ausdruck“ als Wurzel.

*Ausdruck* ::= *⟨Wert⟩*  
              | "(" *⟨Ausdruck⟩* *⟨Binärer Operator⟩* *⟨Ausdruck⟩* ")"  
              | *⟨Unärer Operator⟩* "(" *⟨Ausdruck⟩* ")"

*Wert* ::= "w" | "f"

*Binärer Operator* ::= "and" | "or" | "xor" | "eq"

*Unärer Operator* ::= "not" | "abs"

---

a) (w and f) xor w

3 Punkte

☐ Dieser Text passt

☐ Dieser Text passt nicht

b) not (w eq w)

3 Punkte

☐ Dieser Text passt

☐ Dieser Text passt nicht



### 3. Struktogramm

5 Punkte

Sie liegen im Bett und der Wecker klingelt. Völlig verschlafen werfen Sie einen Blick auf Ihr Smartphone und befolgen den in dieser Aufgabe als Struktogramm zu beschreibenden Algorithmus. Verwenden Sie jede der folgenden Strukturen mindestens einmal: Sequenz, Wiederholung, Alternative, Mehrfachauswahl.

---

Wenn der heutige Tag ein Samstag oder ein Sonntag ist, betätigen Sie wiederholt die „Snooze“-Funktion, bis es 11:00 Uhr ist. Dann stehen Sie auf.

Sollte es jedoch ein normaler Wochentag sein werfen Sie einen Blick auf Ihren Stundenplan. Wenn Sie noch mehr als zwei Stunden haben bis der Unterricht beginnt machen Sie sich ein ausgiebiges englisches Frühstück. Wenn Sie noch zwischen einer und zwei Stunde haben gönnen Sie sich Toast mit Marmelade. Und wenn Sie weniger als eine Stunde haben trinken Sie nur schnell einen Kaffee. In jedem Fall werden Sie sich an einem Wochentag nach dem Frühstück duschen und dann das Haus verlassen.



#### 4. Auswertung von Ausdrücken

Diese Aufgabe beschäftigt sich mit der Auswertung von Ausdrücken in der Programmiersprache Pascal.

##### a) Reihenfolge der Kategorien

2 Punkte

Unten sehen Sie die Tabelle der Operatorenkategorien in Pascal. Tragen Sie in die Spalte „Priorität“ die korrekte Priorität entsprechend der Regeln in Pascal ein.

Operatoren	Kategorie	Priorität
Not, +, -	Unäre Operatoren	
+, -, or, xor	Additive Operatoren	
=, <, <=, >, >=, <>	Relationale Operatoren	
*, /, div, mod, and	Multiplikative Operatoren	

Werten sie die folgenden Ausdrücke gemäß den von Ihnen definierten Prioritäten aus. Führen Sie dafür zunächst Klammern ein und werten Sie diese Klammern in einer jeweils eigenen Zeile aus.

##### b) `2.0 >= 3.0 and not true`

3 Punkte

##### c) `true and true or false or not false`

3 Punkte



## 5. Analyse von PASCAL-Programmen

5 Punkte

Für jedes der im Folgenden zu sehenden Pascal-Programme werden Ihnen mehrere Aussagen und die Anzahl der wahren Aussagen präsentiert. Kreuzen Sie die 5 wahren Aussagen an. Für jedes korrekt gesetzte Kreuz gibt es einen Punkt, sollten Sie mehr als fünf Kreuze machen, erhalten Sie für die Aufgabe 0 Punkte.

**Hinweis:** Das zu sehenden Programm lässt sich fehlerfrei kompilieren.

```
1  program klausur_eins;
2
3  var
4      v1, v2, v3, v4, v5 : Integer;
5
6  begin
7      v4 := 0;
8      v5 := 0;
9
10     readln(v1);
11
12     for v2 := v1 downto 1 do
13     begin
14         readln(v3);
15         v4 += v3;
16         if (v3 > v5) then
17             v5 := v3;
18         end;
19
20     write(v4 / v1, ' ');
21     writeln(v5);
22 end.
```

Dieses Programm ...

- |  |   |
|--|---|
| <input type="checkbox"/> ... gibt in Zeile 20 eine reelle Zahl aus.  | <input type="checkbox"/> ... durchläuft die Schleife in Zeile 12 in Abhängigkeit von dem in Zeile 10 eingegebenen Wert. |
| <input type="checkbox"/> ... gibt in Zeile 21 eine reelle Zahl aus.  | <input type="checkbox"/> ... könnte die Zeilen 20 und 21 auch wie folgt zusammenfassen:<br>writeln(v4 / v1, ' ', v5)    |
| <input type="checkbox"/> ... wertet die Variable v2 innerhalb des Schleifenrumpfes aus.                    | <input type="checkbox"/> ... stürzt ab, wenn der Benutzer eine „0“ eingibt.   |
| <input type="checkbox"/> ... wertet in Zeile 16 einen booleschen Ausdruck aus.                             | <input type="checkbox"/> ... berechnet unter anderem einen minimalen eingegebenen Wert.                                 |
| <input type="checkbox"/> ... liest stets fünf Werte vom Benutzer ein.                                      | <input type="checkbox"/> ... summiert <b>alle</b> vom Benutzer eingegebenen Werte in der Variablen v4 auf.              |
| <input type="checkbox"/> ... gibt für den Benutzer verständliche Anweisungen aus.                          | <input type="checkbox"/> ... speichert den Namen des Anwenders in der Variable v5.                                      |
| <input type="checkbox"/> ... liest in Zeile 17 eine Ganzzahl ein.  |   |
| <input type="checkbox"/> ... durchläuft die Schleife in Zeile 12 bei negativen Eingaben mindestens einmal. |   |



## 6. Schreibtischtest

6 Punkte

```
1  program klausur_eins;
2
3  var
4      v1, v2, v3, v4, v5 : Integer;
5
6  begin
7      v4 := 0;
8      v5 := 0;
9
10     readln(v1);
11
12     for v2 := v1 downto 1 do
13     begin
14         readln(v3);
15         v4 += v3;
16         if (v3 > v5) then
17             v5 := v3;
18         end;
19
20     write(v4 / v1, ' ');
21     writeln(v5);
22 end.
```

Welche Ergebnisse werden für die folgenden Eingaben ausgegeben? Der Schrägstrich steht in diesen Auflis-  
tungen für eine mit <ENTER> bestätigte Eingabe für jeweils eine readln-Anweisung. Sollten bei der Ausgabe  
reelle Zahlen ausgegeben werden, runden Sie diese auf zwei Nachkommastellen.

**Hinweis:** Es handelt sich exakt um das Programm aus Aufgabe 5. Es wird an dieser Stelle ausschließlich auf-  
grund der besseren Lesbarkeit erneut abgedruckt.

Eingabe 1)     1 / 0

Ausgabe 1)

---

Eingabe 2)     5 / 1 / 2 / 3 / 4 / 5

Ausgabe 2)

---

Eingabe 3)     3 / -1 / -1 / -1

Ausgabe 3)

---