

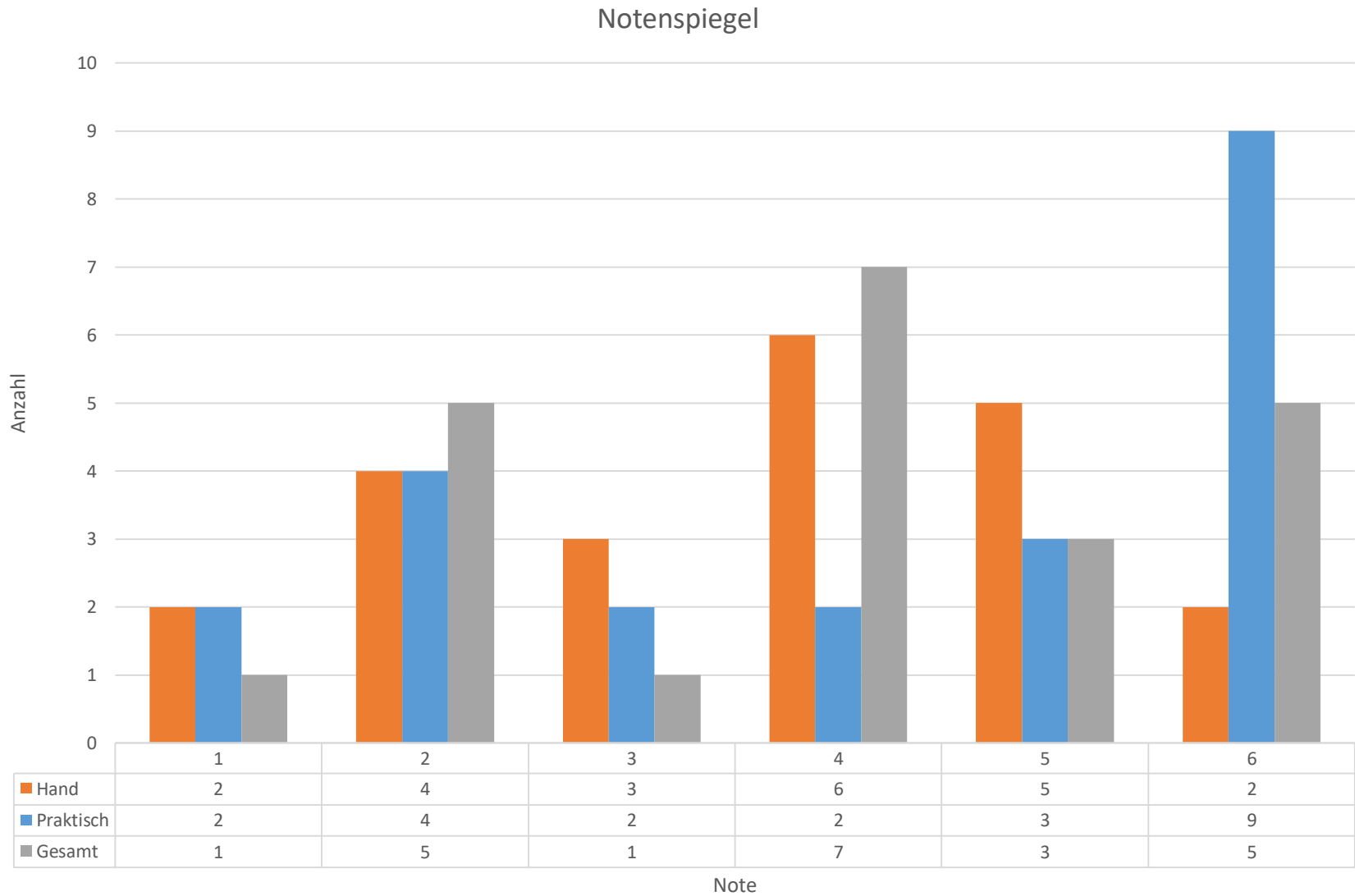
# Programmieren 1



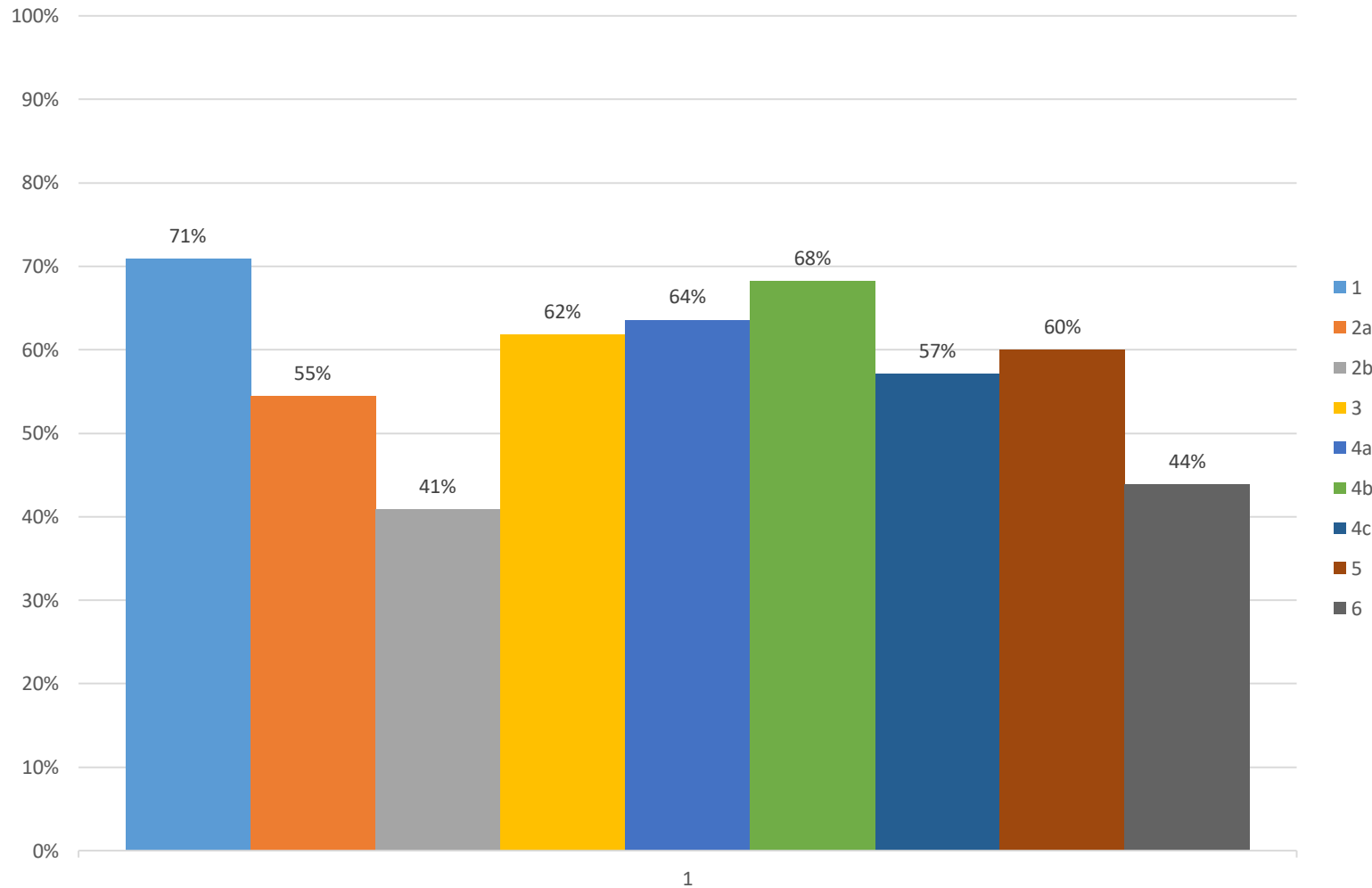
Wintersemester 2016/2017

Marcus Riemer, B.Sc.

Basierend auf den Unterlagen „Programmstrukturen 1“  
von Prof. Dr. Andreas Häuslein



Durchschnittspunkte je Aufgabe





- Beim nächsten mal anders:
  - Kein Bleistift
  - Kein Tackern des Deckblattes
  - Vorgegebene Abkürzungen für längliche Bezeichner, die wiederholt verwendet werden müssen.
  - Ein Bonuspunkt für korrekte Einrückung im praktischen Teil.
- Allgemeine Hinweise zu meiner Korrektur
  - Nicht Häkchen zählen, sondern Abzüge



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.

## Beliebte Fehler

- Falsche Regeldefinition
- Vordefinierte Regel „Buchstabe“ benutzt
- Regel benutzt, aber nicht definiert
- Nichtterminalsymbol vs Terminalsymbol

Nicht richtig, aber auch nicht abgewertet

- Wiederholungen mit Anzahlen

## 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.

|                         |     |  |
|-------------------------|-----|--|
| <i>Ausdruck</i>         | ::= | $\langle \text{Wert} \rangle$<br>  "meee(" $\langle \text{Ausdruck} \rangle$ $\langle \text{Binärer Operator} \rangle$ $\langle \text{Ausdruck} \rangle$ ")"<br>  $\langle \text{Unärer Operator} \rangle$ "(" $\langle \text{Ausdruck} \rangle$ ")" |
| <i>Wert</i>             | ::= | "w"   "f"  |
| <i>Binärer Operator</i> | ::= | "and"   "or"   "xor"   "eq"  |
| <i>Unärer Operator</i>  | ::= | "not"   "abs"  |

### Auszuwertende Texte:

- (w and f) xor w
- not (w eq w)

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.





## Beliebte Fehler

- Es wurde ein Syntaxdiagramm gemalt
- Falsche Schreibweisen für Alternative oder Mehrfachauswahl

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          | 1         |
| +, -, or, xor       | Additive Operatoren       | 3         |
| =, <, <=, >, >=, <> | Relationale Operatoren    | 4         |
| *, /, div, mod, and | Multiplikative Operatoren | 2         |

Noch einen Punkt wenn eines der folgenden Kriterien zutrifft:

- Punkt vor Strichrechnung berücksichtigt wurde
- Mindestens zwei Prioritäten richtig genannt wurden

### Auszuwertende Ausdrücke

- 2.0 >= 3.0 and not true
- true and true or false or not false

```
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 ..?

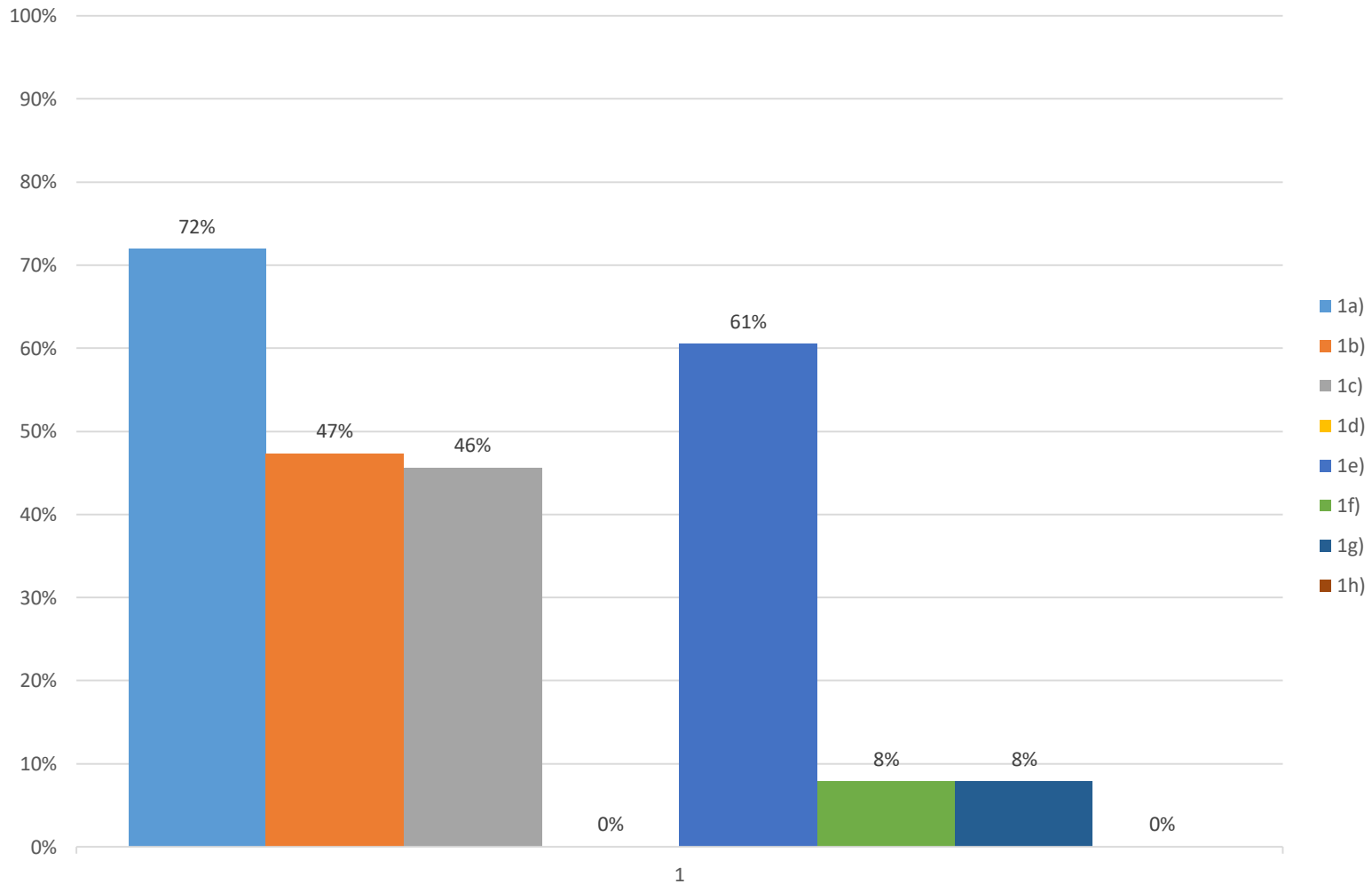
- ... gibt in Zeile 20 eine reelle Zahl aus.
- ... gibt in Zeile 21 eine reelle Zahl aus.
- ... wertet die Variable v2 innerhalb des Schleifenrumpfes aus.
- ... wertet in Zeile 16 einen booleschen Ausdruck aus.
- ... liest stets fünf Werte vom Benutzer ein.
- ... gibt für den Benutzer verständliche Anweisungen aus.
- ... liest in Zeile 17 eine Ganzzahl ein.
- ... durchläuft die Schleife in Zeile 12 bei negativen Eingaben mindestens einmal.

```
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 ..?

- ... durchläuft die Schleife in Zeile 12 in Abhängigkeit von dem in Zeile 10 eingegebenen Wert.
- ... könnte die Zeilen 20 und 21 auch wie folgt zusammenfassen:  
writeln(v4 / v1, ' ', v5)
- ... stürzt ab, wenn der Benutzer eine „0“ eingibt.
- ... berechnet unter anderem einen minimalen eingegebenen Wert.
- ... summiert **alle** vom Benutzer eingegebenen Werte in der Variablen v4 auf.
- ... speichert den Namen des Anwenders in der Variable v5.

Durchschnittspunkte je Aufgabe (Praktisch)



- 3 Teilaufgaben gestrichen
- Keine Abzüge für ...
  - Falsche Formatierung der Ausgabe  
`T_12 = {1;2;3;4;4;6;6;6;6;6;6;12;12;}`
  - Falsche Verwendung der Val-Prozedur  
`wertNeu := val(wert, i, code);`
  - Unsinnige Anweisungen an den Benutzer  
`write('geben Sie Zahl zwischen 1 und und 10 ein: ');`  
`until (teilerMIN > 10)`
  - Vergessene Ausgabe des Teilerintervalls
- Abzüge für ...
  - Wiederholte Angabe von magischen Grenzen im Code  
`writeln('Bitte eine ganze Zahl > 10 eingeben: ');`  
`until ((teilerMin > 10) and (teilerMax > 10));`

- Beliebte Fehler
  - Nichts hochladen!
  - ```
if teilermax < teilerMin then  
    minmax := teilermax;  
    teilermax := teilerMin;  
    teilerMin := minmax;
```

`//warum zur hölle wurde teilermin immer 0???`