

# Unterschiede beim Memorieren von Quelltexten zwischen NovizInnen und ExpertInnen der OOP

Mike Barkmin, Matthias Kramer, David Tobinski und Torsten Brinda

## Hintergrund

- Studie ist Teil des COMMOOP-Projekts [1] zur Erstellung und Validierung eines Kompetenzstrukturmodells der objektorientierten Programmierung
- Ansatz zur Erhebung der Kompetenzdimension "Syntax und Semantik"
- Basierend auf vorherigen Ergebnissen [2] aus dem COMMOOP-Projekt und denen von Adelson [3] wurde die **Hypothese** aufgestellt, dass *ExpertInnen Quelltexte in semantischen Einheiten im Arbeitsgedächtnis ablegen, wohingegen NovizInnen syntaktische Einheiten verwenden.*

```

1 2 Program 1
1 [1.0] [x] $sort.within(sorting.list) i, j, temp.sort
2 [1.1] [x] for i <- 1:length(sorting.list) do thru %3
3 [1.2] [x] for j <- i:length(sorting.list) do thru %3
4 [1.3] [x] if sorting.list[i] > sorting.list[j] then,
temp.sort <- sorting.list[j];
sorting.list[j] <- sorting.list[i];
sorting.list[i] <- temp.sort
5 [1.4] [x] return sorting.list

Program 2
6 [2.0] [x] $random.to.new(old.list) new.list, n, random.place
7 [2.1] [x] new.list <- make(tuple, length(old.list), null); n <- 0
8 [2.2] [x] loop: n <- n + 1; if n = length(old.list) + 1 then
return new.list
9 [2.3] [x] random.place <- int(length(old.list) * random(0)) + 1
10 [2.4] [x] if old.list[random.place] # null then
new.list[n] <- old.list[random.place];
old.list[random.place] <- null;
goto loop else goto %3
    
```



## Auswertung / Ergebnisse

• 42 Studierende (5 unvollständige Datensätze; 10 Frauen; 27 Männer; Altersdurchschnitt: 25.03, SD: 3.77): 31 Studierende eines Studiengangs mit informatischem Fokus; 11 aus anderen Studiengängen

• Protokollierte Tastenanschläge und Merkphasen wurden genutzt, um Unterschiede zwischen NovizInnen und ExpertInnen zu ermitteln.

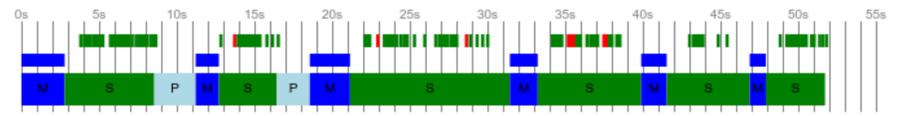


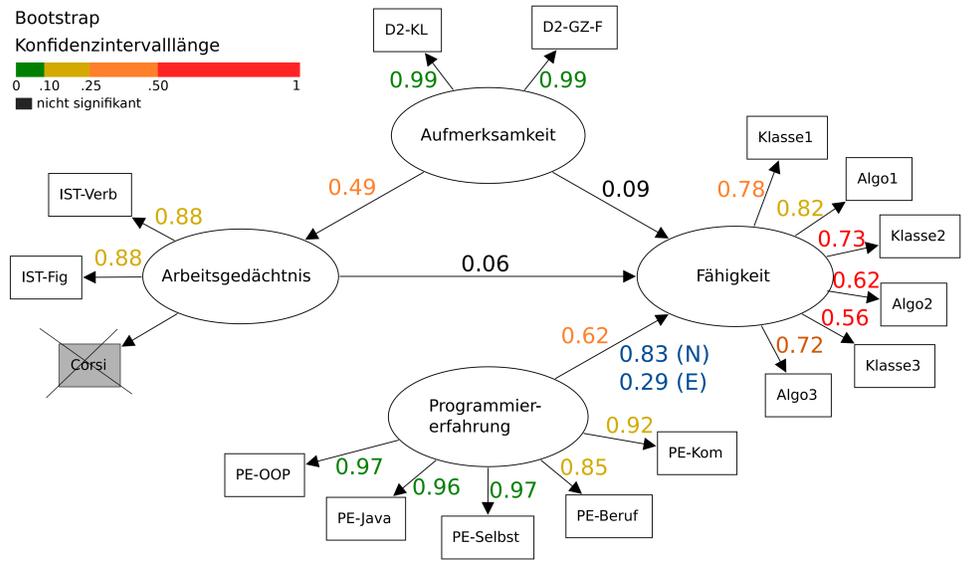
Abb: Zeitstrahl eines Bearbeitungsprozesses  
1. Zeile: Tastenanschläge (rot = Backspace oder Delete), 2. Zeile: Merkphasen  
3. Zeile: Einteilung in Merkphasen (M), Schreibphasen (S) und Pausen (P)

• Keine signifikanten Unterschiede in den natürlichsprachlichen Items => keine unterschiedlichen Grundvoraussetzungen

• Überprüfung eines aus der Theorie abgeleiteten Strukturgleichungsmodells mittels PLSPM [8] hat gezeigt, dass die Programmiererfahrung die Fähigkeit beim Memorieren und Reproduzieren von Quelltexten am besten erklärt.

• Bootstrapping-Validierung mit 5000 Samples ergab, dass nur der Einfluss der Programmiererfahrung auf die Fähigkeit und der Einfluss der Aufmerksamkeit auf das Arbeitsgedächtnis signifikant sind.

=> Keine Beeinflussung des Ergebnisses durch kognitive Unterschiede der Testpersonen.



## Vorstellung des Designs

• Erhebung von kognitiven Fähigkeiten, um Einfluss auf den Bearbeitungsprozess bewerten zu können

• Erhebung der Fähigkeit, Quelltexte und natürlichsprachliche Texte zu memorieren und zu reproduzieren

Drücke, um den Text anzeigen zu lassen

58.33%

```

class Datei {
String name;
String pfad;

Datei(String name, String pfad) {
this.name = name;
}
}
    
```

Alle Aktionen der Testpersonen wurden aufgezeichnet.

Weiter

• Erhebung von sozio-demographischen Angaben und der selbst eingeschätzten Programmiererfahrung in Anlehnung an [7]

- Wie schätzen Sie Ihre Programmiererfahrung ein?
- Wie erfahren sind Sie mit der Programmiersprache Java?
- Wie erfahren sind Sie mit dem Paradigma Objektorientierte Programmierung?
- Wie schätzen Sie Ihre Programmiererfahrung verglichen mit ExpertInnen mit 20-jähriger Berufserfahrung ein?
- Wie schätzen Sie Ihre Programmiererfahrung verglichen mit Ihren KommilitonInnen ein?

Wie erfahren sind Sie mit dem Paradigma Objektorientierte Programmierung?

1 2 3 4 5 6 7 8 9 10

1 = unerfahren, 10 = sehr erfahren

## Fazit und Ausblick

• Die Ergebnisse aus vorherigen Forschungen [3, 9, 10, 11, 12, 13] konnten für die objektorientierte Programmiersprache Java bestätigt werden.

• Gerade im Anfängerbereich sind deutliche Unterschiede beim Merken und Reproduzieren von Quelltexten zu erkennen.

• Bevor das Messinstrument zur Kompetenzmessung verwendet werden kann, müssen stärker differenzierende Items entwickelt werden.

• Zwecks Optimierung sollten noch weitere Varianten erprobt werden, zum Beispiel die Beschränkung der Bearbeitungszeit und / oder die Beschränkung der Merkphasen.

• In zukünftigen Forschungen könnte untersucht werden, ob die Kompetenz in den Dimensionen Syntax und Semantik auch auf andere Programmiersprachen übertragbar ist.

Weitere Details zur Durchführung und Auswertung sind in [14] zu finden.

## Referenzen

[1] Kramer, M., P. Hubwieser und T. Brinda (2016). „A Competency Structure Model of Object-Oriented Programming“. In: 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE), S. 1–8.

[2] Kramer, M., D. A. Tobinski und T. Brinda (2016). „On the Way to a Test Instrument for Object-Oriented Programming Competencies“. In: Proceedings of the 16th Koli Calling International Conference on Computing Education Research, Koli Calling '16, Koli, Finland, ACM, S. 145–149.

[3] Adelson, B. (1981). „Problem solving and the development of abstract categories in programming languages“. In: Memory & Cognition 9, 4, S. 422–433.

[4] Brickenkamp, R. (1994). Test d2: Aufmerksamkeits-Belastungs-Test. Bd. 8. Göttingen: Hogrefe.

[5] Corsi, P. M. (1972). Human Memory and the Medial Temporal Region of the Brain. McGill theses.

[6] Amthauer, R. u. a. (2006). Intelligenz-Struktur-Test 2000 R. Bd. 2. Göttingen: Hogrefe.

[7] Siegmund, J. u. a. (2014). „Measuring and Modeling Programming Experience“. In: Empirical Softw. Engg. 19, 5, S. 1299–1334.

[8] Hair, J. F. (2013). A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM).

[9] Shneiderman, B. (1976). „Exploratory experiments in programmer behavior“. In: International Journal of Computer & Information Sciences 5, 2, S. 123–143.

[10] Barfield, W. (1986). „Expert-novice differences for software: implications for problem-solving and knowledge acquisition“. In: Behaviour & Information Technology 5, 1, S. 15–29.

[11] Schmidt, A. L. (1986). „Effects of experience and comprehension on reading time and memory for computer programs“. In: International Journal of Man-Machine Studies 25, 4, S. 399–409.

[12] Bateson, A. G., R. A. Alexander und M. D. Murphy (1987). „Cognitive processing differences between novice and expert computer programmers“. In: International Journal of Man-Machine Studies 26, 6, S. 649–660.

[13] Magliaro, S. und J. K. Burtin (1987). „Adolescents' Chunking of Computer Programs“. In: Comput. Sch. 4, 3-4, S. 129–138.

[14] Barkmin, M. (2017). Konstruktion und Erprobung eines Bausteins zur Kompetenzmessung im Bereich der objektorientierten Programmierung im den Dimensionen Syntax und Semantik. Universität Duisburg-Essen.

## Kontakt

Mike Barkmin, Matthias Kramer  
und Torsten Brinda  
Didaktik der Informatik  
Universität Duisburg-Essen  
mike.barkmin@uni-due.de  
matthias.kramer@uni-due.de  
torsten.brinda@uni-due.de

David Tobinski  
Institut für Psychologie  
Universität Duisburg-Essen  
david.tobinski@uni-due.de

