



UNIVERSITÄT  
DUISBURG  
ESSEN

*Offen im Denken*

## **Automatische Auswertung von Leistungserhebungen zur unmittelbaren Diagnostik im (Informatik-)Unterricht**

Kolloquium Institut für Psychologie Jena

Mike Barkmin & Matthias Kramer ■ 15. Januar 2020

# Inhaltsverzeichnis

1. Einleitung
2. Motivation
3. Vorbetrachtungen
4. Das Online-Assessment-Tool
5. Zusammenfassung
6. Ausblick



Abbildung: Bild von TeroVesalainen unter Pixabay License via Pixabay

# Einleitung

# Wer sind wir?



Matthias Kramer



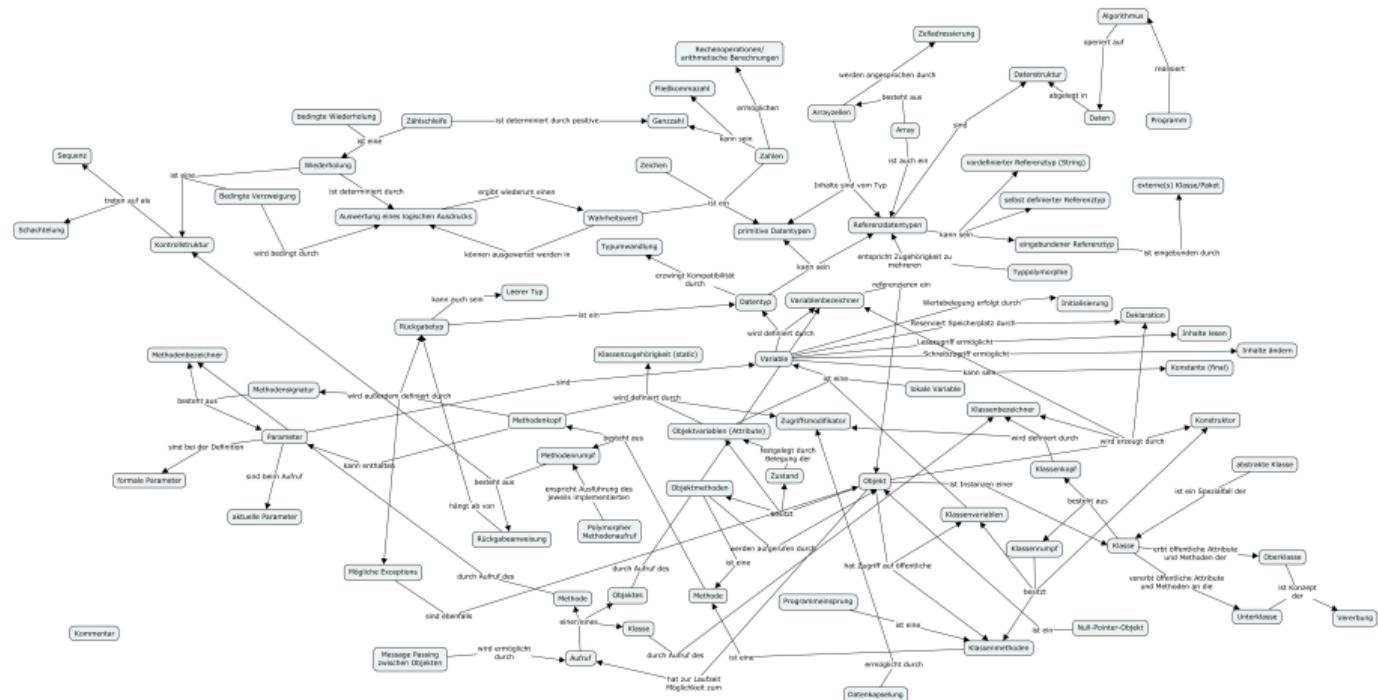
Lehrstuhl für Didaktik der Informatik

Universität Duisburg-Essen



Mike Barkmin

## Woran forschen wir?



# Motivation

# Motivation



Abbildung: Bild von Ag Ku unter Pixabay License via Pixabay

# Motivation

- Bei Erhebungen haben wir Probleme festgestellt, die so auch schon in der Literatur beschrieben wurden



Abbildung: Bild von Ag Ku unter Pixabay License via Pixabay

# Motivation

- Bei Erhebungen haben wir Probleme festgestellt, die so auch schon in der Literatur beschrieben wurden
- Zeitaufwand bei der Digitalisierung und späterer Auswertung



Abbildung: Bild von Ag Ku unter Pixabay License via Pixabay

# Motivation

- Bei Erhebungen haben wir Probleme festgestellt, die so auch schon in der Literatur beschrieben wurden
- Zeitaufwand bei der Digitalisierung und späterer Auswertung
- Größere Stichproben sehr aufwändig zu bewerkstelligen



Abbildung: Bild von Ag Ku unter Pixabay License via Pixabay

# Motivation

- Bei Erhebungen haben wir Probleme festgestellt, die so auch schon in der Literatur beschrieben wurden
- Zeitaufwand bei der Digitalisierung und späterer Auswertung
- Größere Stichproben sehr aufwändig zu bewerkstelligen
- Komplexere aber realistischere Aufgabentypen schwer zu realisieren (Stichwort: Programmieren auf Papier)



Abbildung: Bild von Ag Ku unter Pixabay License via Pixabay

# Vorberachtungen

- Webapplikation (WA)
- Vorerst keine Notwendigkeit für Nutzeraccount ⇒ Zugang zum Test nur durch Besuch einer Webseite & Token (NUA)
- Analyse der Problemlösefähigkeiten durch Untersuchung der Nutzerinteraktion mit den Aufgaben ⇒ Tracken der Nutzerinteraktion mit den Items (Anzahl der Versuche, Zeit, Reihenfolge, Lösungsansätze etc.) (UIT)
- Datenschutz: entweder verschlüsselte Ablage der Daten oder mind. Server an deutschen Hochschulen (DS)
- Möglichkeiten zur selbstständigen Item- und Testerstellung (ITE)
- Evtl. Erweiterbarkeit um neue Item-Formate ⇒ direkte Ansprachemöglichkeit (CE)
- Möglichkeit zum Herunterladen der Daten in bspw. CSV-Format zur Weiterverarbeitung durch vorgefertigte R-Skripte oder bereits serverseitige Verarbeitung

|                                   | WA  | NUA | UIT | DS | ITE | CE |
|-----------------------------------|-----|-----|-----|----|-----|----|
| JACK (Goedicke und Striewe, 2017) | (✓) | ✗   | ✗   | ✓  | ✓   | ✓  |
| VILLE (Rajala u. a., 2016)        | ✓   | ✗   | ?   | ✓  | ✓   | ✓  |
| TRAKLA2 (Laakso u. a., 2004)      | (✓) | ✗   | ✓   | ✓  | ?   | ?  |
| BOSS2 (Joy u. a., 2005)           | (✓) | ✗   | ✗   | ✓  | ?   | ?  |
| ProGoSS (Gluga u. a., 2011)       | (✓) | ✗   | ✗   | ✗  | ?   | ?  |
| QuizJET (Hsiao u. a., 2008)       | ✓   | ✓   | ✗   | ✗  | ?   | ?  |

- Zusätzlich wurden weitere Systeme (Mooshak, Bottlenose, CourseMarker, WeBWorK uvm.) betrachtet, keines für unsere Zwecke schnell einsetzbar
- ⇒ Eigene Systementwicklung war notwendig

# Das Online-Assessment-Tool

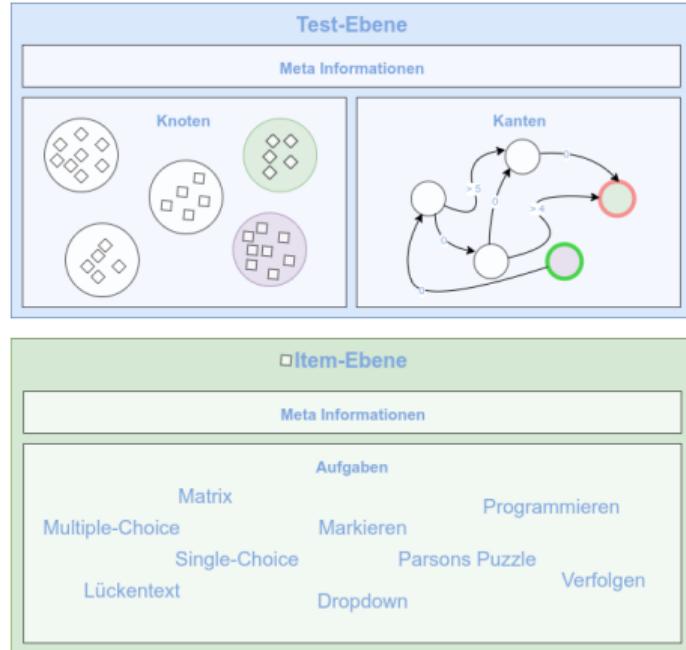
# Das Online-Assessment-Tool

1. Einleitung
2. Motivation
3. Vorbetrachtungen
4. Das Online-Assessment-Tool
  - 4.1 Konzeptionelle Realisierung
  - 4.2 Technische Realisierung
  - 4.3 Item-Ebene
  - 4.4 Test-Ebene
5. Zusammenfassung
6. Ausblick



Abbildung: Bild von TeroVesalainen unter Pixabay License via Pixabay

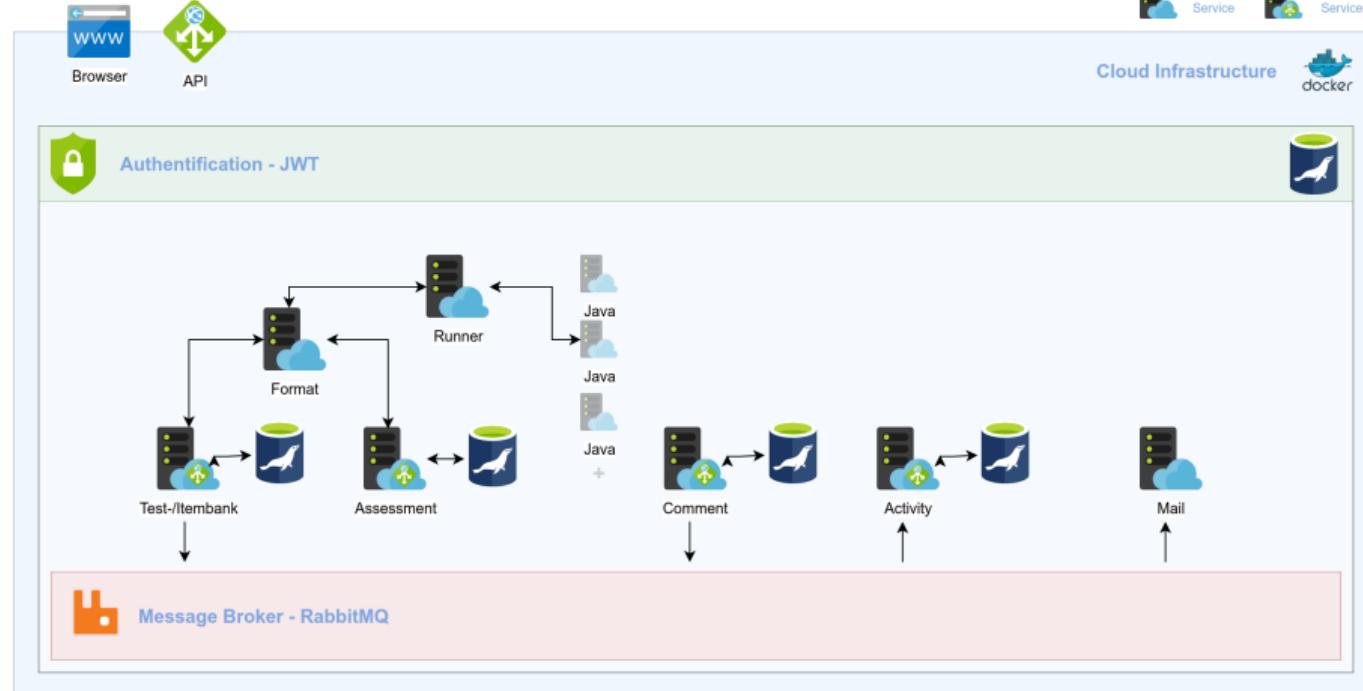
# Konzeptionelle Realisierung



# Technische Realisierung



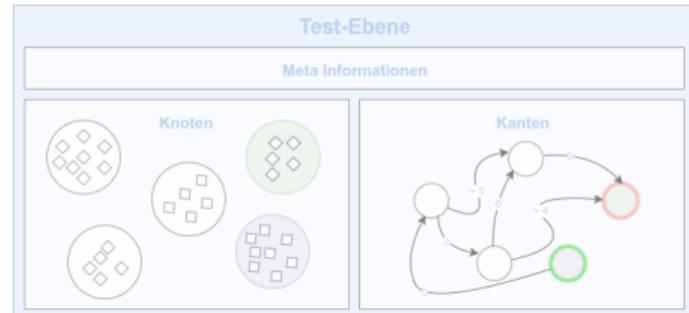
Cloud Infrastructure



## 4. Das Online-Assessment-Tool

- 4.1 Konzeptionelle Realisierung
- 4.2 Technische Realisierung
- 4.3 Item-Ebene**
  - 4.3.1 Analog zu Digital
  - 4.3.2 Komplexere Authentische Aufgabentypen
  - 4.3.3 Betrachtung des Lösungswegs

## 4.4 Test-Ebene



# Analog zu Digital I

## ■ Single/Multiple-Choice

- Analoge Aufgabentypen digitalisieren
- Dadurch schnellere Auswertung
- Direkte Visualisierung
- (Versehentliches Vergessen einer Antwort kann abgefangen werden)

**Aufgabe**

**Selbsteinschätzung**  
Bitte bearbeiten Sie die untenstehende Aufgabe.

Bitte schätzen Sie Ihre Fähigkeiten auf einer vierstufigen Skala ein:

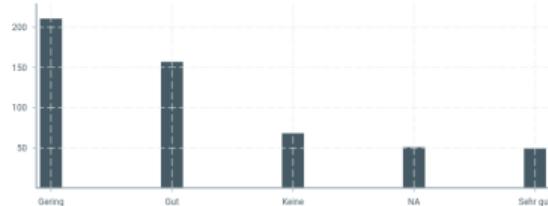
|   | Keine                 | Gering                | Gut                   | Sehr gut              |
|---|-----------------------|-----------------------|-----------------------|-----------------------|
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein:     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren allgemein wie folgt ein:    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Ich schätze mein Wissen im Bereich der Informatik unabhängig vom Programmieren wie folgt ein: | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

536

Ich schätze mein Wissen im Bereich der Informatik unabhängig vom Programmieren wie folgt ein:

Ich schätze meine Fähigkeiten im objektorientierten Programmieren allgemein wie folgt ein:

Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein:



# Analog zu Digital II

## ■ Lückentext

- „Handschrifterkennung“ entfällt
- Auswertung kann direkt durch reguläre Ausdrücke realisiert werden
- Diagnostische Visualisierung als Wortwolke
- Siehe Striewe u. a. (2017)

Aufgabe

Sie sehen hier den Inhalt der Datei Aba.java. Welches Java-Schlüsselwort fehlt, damit diese Datei fehlerfrei kompiliert bzw. übersetzt wird?

```
Aba{  
    }  
}
```



# Komplexere Authentische Aufgabentypen

- Wünschenswert auch authentische Aufgabentypen einsetzen zu können
- Dazu haben wir einen Runner implementiert
- Quelltext wird bei uns auf dem Server kompiliert und gegen Unitests getestet

```
1. import java.util.Scanner;
2.
3. public class HelloWorld {
4.
5.     public static void main(String[] args) {
6.
7.         // Creates a reader instance which takes
8.         // input from standard input - keyboard
9.         Scanner reader = new Scanner(System.in);
10.        System.out.print("Enter a number: ");
11.
12.        // nextInt() reads the next integer from the keyboard
13.        int number = reader.nextInt();
14.
15.        // println() prints the following line to the output screen
16.        System.out.println("You entered: " + number);
17.    }
18. }
```

# Entwicklung Komplexerer Aufgabentypen: Highlighting I

- Vergleichsweise einfacher Itemtyp, trotzdem realistisch
- Verbindet konzeptuelles Wissen mit Repräsentation in formalen Sprachen
- Idee: Markiere alle Vorkommen von <Konzept> im gegebenen Quelltext
- Notwendige Aktionen: Verändern des CSS-Attributs (Hintergrundfarbe), Speichern der Indizes

## Aufgabe

### Klassenbezeichner

Als Klassenbezeichner werden alle Worte bezeichnet, die auf eine Java-Klasse verweisen. Bitte markieren Sie im folgenden Quelltextfragment **alle auftretenden Klassenbezeichner**, auch wenn sie doppelt vorkommen.

```
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private TimerThread t;

    private void start()
    {
        clockRunning = true;
    }

    private void stop()
    {
        clockRunning = false;
    }

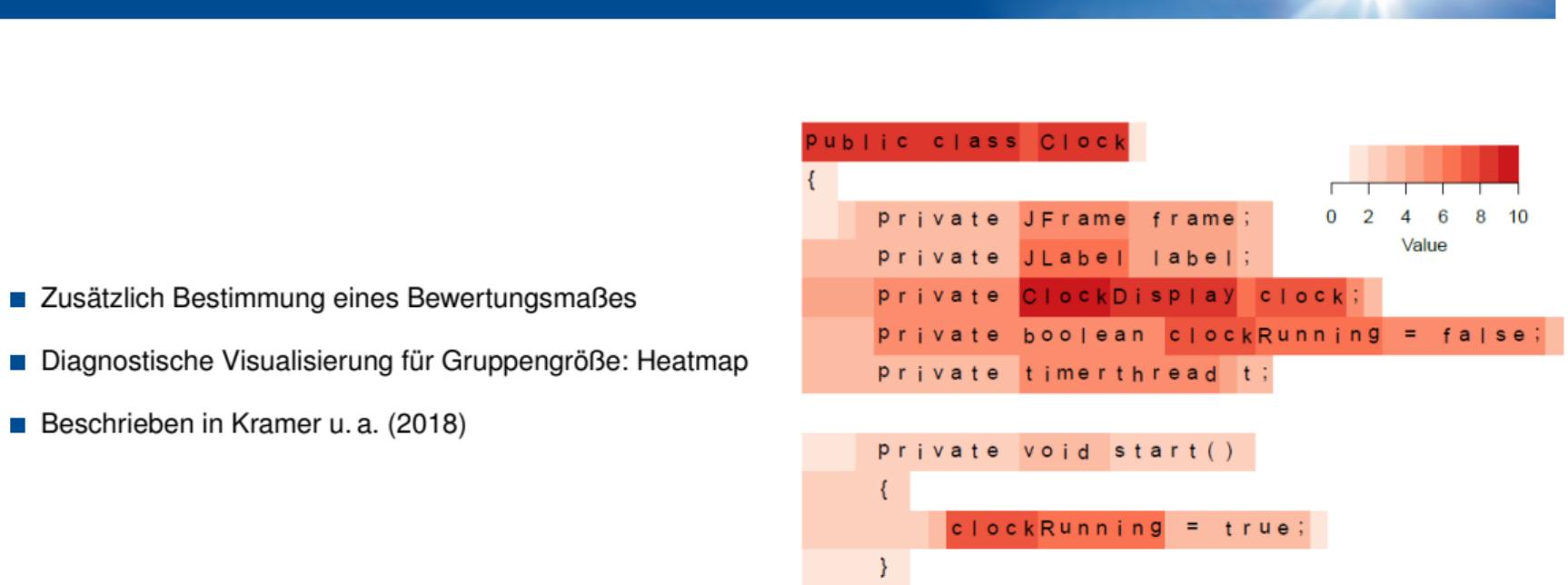
    private void step()
    {
        clock.timeTick();
        label.setText(clock.getTime());
    }

    private void makeMenuBar(JFrame frame)
    {
        final int SHORTCUT_MASK =
            Toolkit.getDefaultToolkit().getMenuShortcutKeyMask();

        JMenuBar menubar = null;
        frame.setJMenuBar(menubar);
    }
}
```

[ALLE MARKIERUNGEN LÖSCHEN](#)

# Entwicklung Komplexerer Aufgabentypen: Highlighting II



- Zusätzlich Bestimmung eines Bewertungsmaßes
- Diagnostische Visualisierung für Gruppengröße: Heatmap
- Beschrieben in Kramer u. a. (2018)

## Betrachtung des Lösungswegs

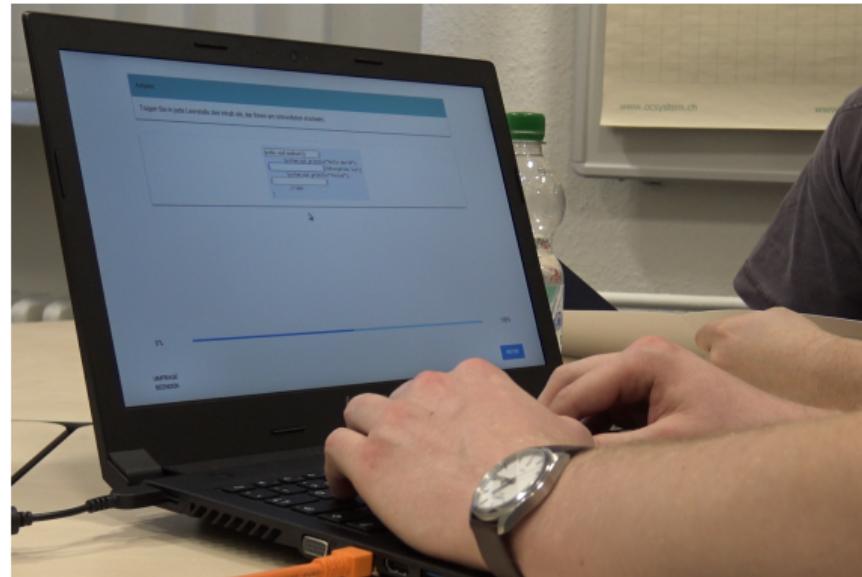
- Durch die Betrachtung des Endergebnisses gehen wichtige Information verloren
- Der Lösungsweg könnte zusätzlich betrachtet werden



Abbildung: Bild von Bhuvanesh S unter Pixabay License via Pixabay

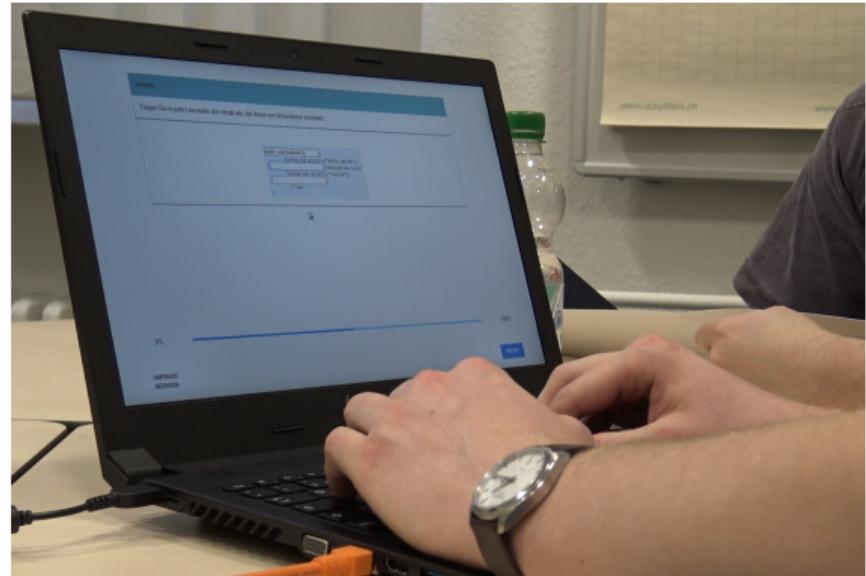
# Betrachtung des Lösungswegs

- Durch die Betrachtung des Endergebnisses gehen wichtige Information verloren
- Der Lösungsweg könnte zusätzlich betrachtet werden
- Lösung: Videoaufzeichnung in HD der Bearbeitung
  - 8 Probanden ca. 30 min ~ 140GB



# Betrachtung des Lösungswegs

- Durch die Betrachtung des Endergebnisses gehen wichtige Information verloren
- Der Lösungsweg könnte zusätzlich betrachtet werden
- Lösung: Videoaufzeichnung in HD der Bearbeitung
  - 8 Probanden ca. 30 min ~ 140GB
- Alternative Lösung: Aufzeichnung der Interaktionen mit dem Assessment-Tool
  - ca. 500 Probanden ~ 20MB



# Aufzeichnen des Lösungswegs

- Action: Ist eine Aktion die vom Benutzer ausgeführt wird
- Reducer: Entscheidet wie sich der Store abhängig von der Aktion ändern soll
- Store: Eine Sammlung von Daten
- UI: Wird abhängig vom Store dargestellt

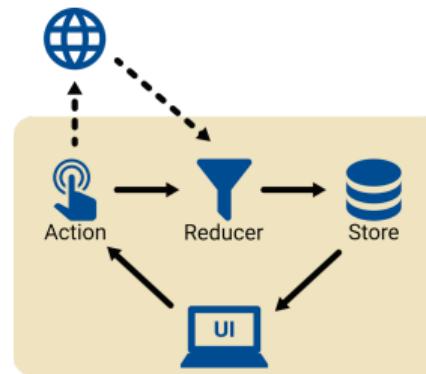


Abbildung: Action-Reducer-Store angelehnt an <https://redux.js.org>

# Aufzeichnung des Lösungswegs - Beispiel

- Jeder Aufgabentyp funktioniert nach dem gleichen Prinzip
- Jeder Aufgabentyp hat eigens definierte Actions
- Actions des Matrix-Single-Choice Aufgabentyps
  - CHECK\_CHOICE

Bitte schätzen Sie Ihre Fähigkeiten auf einer vierstufigen Skala ein:

|   | Keine                 | Gering                | Gut                   | Sehr gut              |
|---|-----------------------|-----------------------|-----------------------|-----------------------|
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein:     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren allgemein wie folgt ein:    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Ich schätze mein Wissen im Bereich der Informatik unabhängig vom Programmieren wie folgt ein: | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Matrix-Single-Choice Aufgabentyp

# Aufzeichnung des Lösungswegs - Beispiel

| Action       | Data   | Time (in ms) |
|--------------|--------|--------------|
| CHECK_CHOICE | [0, 2] | 1100         |

Bitte schätzen Sie Ihre Fähigkeiten auf einer vierstufigen Skala ein:

|   | Keine                 | Gering                | Gut                              | Sehr gut              |
|---|-----------------------|-----------------------|----------------------------------|-----------------------|
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein:     | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren allgemein wie folgt ein:    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/> |
| Ich schätze mein Wissen im Bereich der Informatik unabhängig vom Programmieren wie folgt ein: | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/> |

Matrix-Single-Choice Aufgabentyp

# Aufzeichnung des Lösungswegs - Beispiel

| Action       | Data   | Time (in ms) |
|--------------|--------|--------------|
| CHECK_CHOICE | [0, 2] | 1100         |
| CHECK_CHOICE | [1, 2] | 2000         |

Bitte schätzen Sie Ihre Fähigkeiten auf einer vierstufigen Skala ein:

|   | Keine                 | Gering                | Gut                              | Sehr gut              |
|---|-----------------------|-----------------------|----------------------------------|-----------------------|
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein:     | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Ich schätze meine Fähigkeiten im objektorientierten Programmieren allgemein wie folgt ein:    | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Ich schätze mein Wissen im Bereich der Informatik unabhängig vom Programmieren wie folgt ein: | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/> |

Matrix-Single-Choice Aufgabentyp

# Nutzung des Lösungswegs für Visualisierungen

# Nutzung des Lösungswegs für Visualisierungen - Parsons Puzzle

Aufgabe

### Parsons Puzzles

Selbstverständlich ist man nicht auf Programmierfragmente beschränkt. Außerdem gibt es noch weitere Optionen. So kann man bspw. forcieren, dass von einer Menge von Fragmenten stets nur eins ausgewählt werden kann. Oder dass bestimmte Elemente immer in einer festen Reihenfolge vorkommen müssen.

The screenshot shows a 'Drag from here' column on the left and a 'Drop here' column on the right. In the 'Drop here' column, there are two boxes: 'Ein Modell aus der Item-Response-Theorie ist das' (with a count of 1) and 'Das Besondere daran ist,' (with a count of 2). In the 'Drag from here' column, there are five items with counts: 'Birnbaum-Modell.' (count 1), 'Rasch-Modell.' (count 1), 'Partial-Credit-Modell.' (count 1), 'Graded-Response-Modell.' (count 1), and a multi-line text block about dichotomous items (count 2).

Initialer Zustand

- Basiert auf der Arbeit von Parsons und Haden (2006)

### Actions für Parsons Puzzles

- MOVE\_FROM\_SOURCE\_TO\_USER (sourceId, userId)
- MOVE\_FROM\_USER\_TO\_SOURCE (userId, sourceId)
- MOVE\_WITHIN\_USER (userId1, userId2)

### Was passiert wenn die Action

MOVE\_FROM\_SOURCE\_TO\_USER (0, 1) ausgeführt wird?

# Nutzung des Lösungswegs für Visualisierungen - Parsons Puzzle

**Aufgabe**

### Parsons Puzzles

Selbstverständlich ist man nicht auf Programmierfragmente beschränkt. Außerdem gibt es noch weitere Optionen. So kann man bspw. forcieren, dass von einer Menge von Fragmenten stets nur eins ausgewählt werden kann. Oder dass bestimmte Elemente immer in einer festen Reihenfolge vorkommen müssen.

Drag from here

- Birnbaum-Modell. ≈1
- dass damit auch Items mit mehreren Ausprägungsstufen und unterschiedlichen Trennschärfen hinsichtlich Personenfähigkeit und Itemschwierigkeit untersucht werden können. ≈2
- Rasch-Modell. ≈1
- dass es bei dichotomen Items und angenommenen gleichen Trennschärfen, die Trennung von Personenfähigkeit und Itemschwierigkeit erlaubt. ≈2
- Partial-Credit-Modell. ≈1
- Graded-Response-Modell. ≈1

Drop here

- Ein Modell aus der Item-Response-Theorie ist das ≈1
- Das Besondere daran ist, ≈2

Initialer Zustand

**Aufgabe**

### Parsons Puzzles

Selbstverständlich ist man nicht auf Programmierfragmente beschränkt. Außerdem gibt es noch weitere Optionen. So kann man bspw. forcieren, dass von einer Menge von Fragmenten stets nur eins ausgewählt werden kann. Oder dass bestimmte Elemente immer in einer festen Reihenfolge vorkommen müssen.

Drag from here

- Birnbaum-Modell. ≈1
- Rasch-Modell. ≈1
- dass es bei dichotomen Items und angenommenen gleichen Trennschärfen, die Trennung von Personenfähigkeit und Itemschwierigkeit erlaubt. ≈2
- Partial-Credit-Modell. ≈1
- Graded-Response-Modell. ≈1

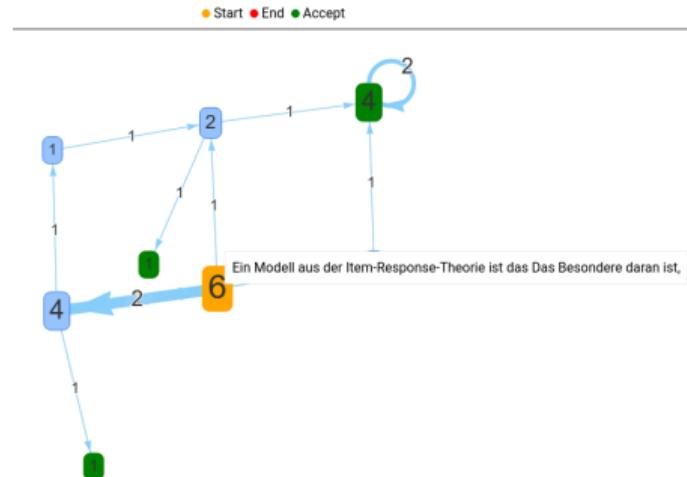
Drop here

- Ein Modell aus der Item-Response-Theorie ist das ≈1
- Das Besondere daran ist, ≈2
- dass damit auch Items mit mehreren Ausprägungsstufen und unterschiedlichen Trennschärfen hinsichtlich Personenfähigkeit und Itemschwierigkeit untersucht werden können. ≈2

Nach Ausführung der Action MOVE\_FROM\_SOURCE\_TO\_USER (0, 1)

# Nutzung des Lösungswegs für Visualisierungen - Parsons Puzzle

- Die Ausführung einer Action führt zu einer Zustandsänderung der Daten
- Jeder Zustand des Parsons-Puzzle ist durch einen Knoten symbolisiert
- Jeder Kante ist die Ausführung einer Action
- Die Zahl/Dicke der Kanten/Knoten symbolisiert wie häufig sie erreicht wurden



Visualisierung von 6 Lösungswegen angelehnt an Helminen u. a. (2012)

# Nutzung des Lösungswegs zum Aufdecken kognitiver Strukturen - Quelltext memorieren (Barkmin u. a., 2017)

Drücke, um den Text anzeigen zu lassen

8.02%

```
class Datei {  
}
```

Weiter

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

# Nutzung des Lösungswegs zum Aufdecken kognitiver Strukturen - Quelltext memorieren

Drücke, um den Text anzeigen zu lassen

0%

Abschicken

## ■ Actions für Quelltext memorieren

- INSERT\_CHAR (charId, pos)
- REMOVE\_CHAR (pos)
- OPEN\_SOURCE\_CODE
- CLOSE\_SOURCE\_CODE

# Nutzung des Lösungswegs zum Aufdecken kognitiver Strukturen - Quelltext memorieren

- Viele Aktionen können unübersichtlich werden
- Es wird jeder Tastendruck aufgezeichnet
- Daher müssen Aktionen zusammengefasst werden
- Hier in Merkphasen (Blau), Schreibphasen (Grün) und Pausephasen (Hellblau)

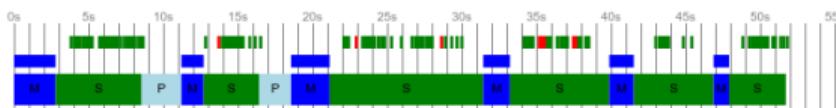


Abbildung: Zeitleiste einer Bearbeitung

| Phase | Zeit in s   | LS  | G | Inhalt   |
|-------|-------------|-----|---|--|
| 5 M   | 2.50        |     |   |  |
| 6 S   | 5.71        | -33 | 0 | uuuupublic <u>class</u> Haus{<br>uuuuprivate <u>int</u> nummer;<br>uuuuprivate <u>String</u> uStringu<br>uuuu                    |
| 7 P   | 2.43        |     |   |  |
| 8 S   | 3.01        | -6  | 1 | uuuupublic <u>class</u> Haus{<br>uuuuprivate <u>int</u> nummer;<br>uuuuprivate <u>String</u> uStraße;<br>uuuu<br>uuuu            |
| 9 P   | 2.43        |     |   |  |
|       | 8.72 (4.86) | -39 | 1 |  |
| 10 M  | 2.15        |     |   |  |
| 11 P  | 2.19        |     |   |  |
| 12 S  | 5.67        | 0   | 8 | uuuupublic <u>class</u> Haus{<br>uuuuprivate <u>int</u> nummer;<br>uuuuprivate <u>String</u> uStfaragbe;<br>uuuu<br>uuuu<br>uuuu |

Abbildung: Transkript einer Bearbeitung

## Auch für psychologische Tests nutzbar

- Digitale Version der Corsi-Blöcke (Corsi, 1972)
- Separat entwickelt (Offline-Anwendung für Mac, Windows und Linux)
- Blöcke können beliebig angeordnet werden
- Basiert auch auf Action-Reducer-Store
- Actions
  - MOUSE\_MOVED (x, y)
  - CORNER\_PRESSED
  - CORNER\_RELEASED
  - BLOCK\_PRESSED (id)
  - BLOCK\_RELEASED (id)

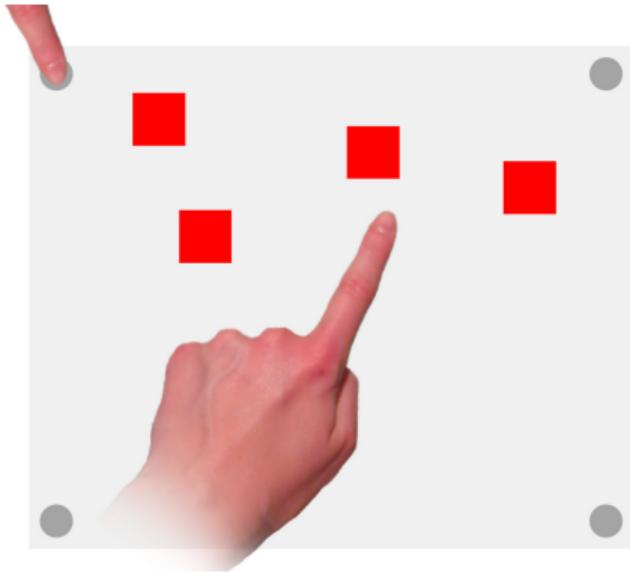


Abbildung: Digitale Version der Corsi-Blöcke  
{mike.barkmin, matthias.kramer}@uni-due.de  
Automatisches Online-Assessment

## Auch für psychologische Tests nutzbar

- Digitale Version der Corsi-Blöcke (Corsi, 1972)
- Separat entwickelt (Offline-Anwendung für Mac, Windows und Linux)
- Blöcke können beliebig angeordnet werden
- Basiert auch auf Action-Reducer-Store
- Actions
  - MOUSE\_MOVED (x, y)
  - CORNER\_PRESSED
  - CORNER\_RELEASED
  - BLOCK\_PRESSED (id)
  - BLOCK\_RELEASED (id)

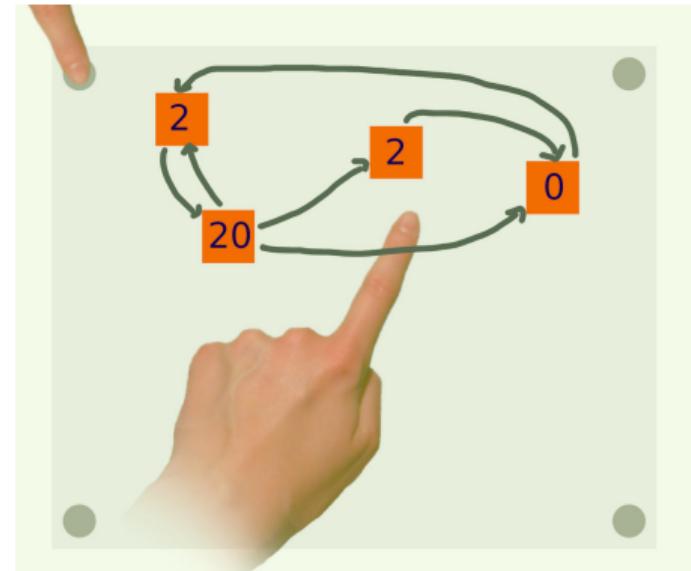


Abbildung: Digitale Version der Corsi-Blöcke

## 4. Das Online-Assessment-Tool

4.1 Konzeptionelle Realisierung

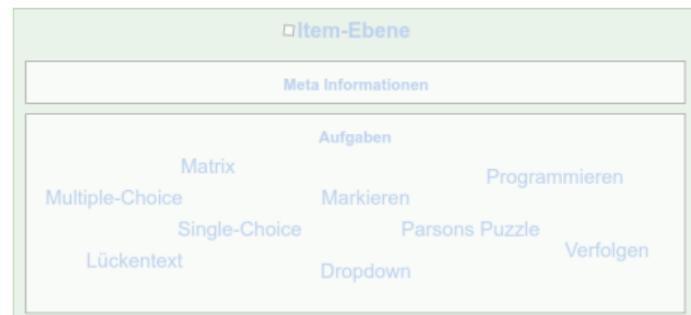
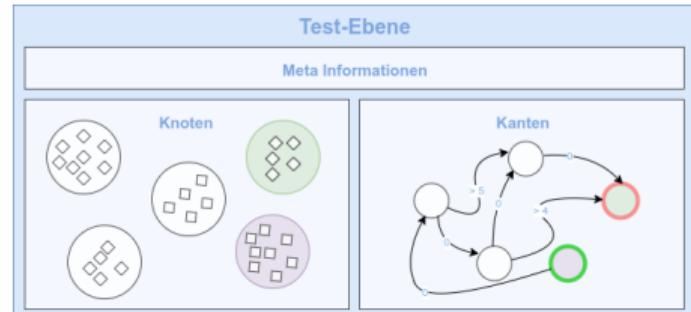
4.2 Technische Realisierung

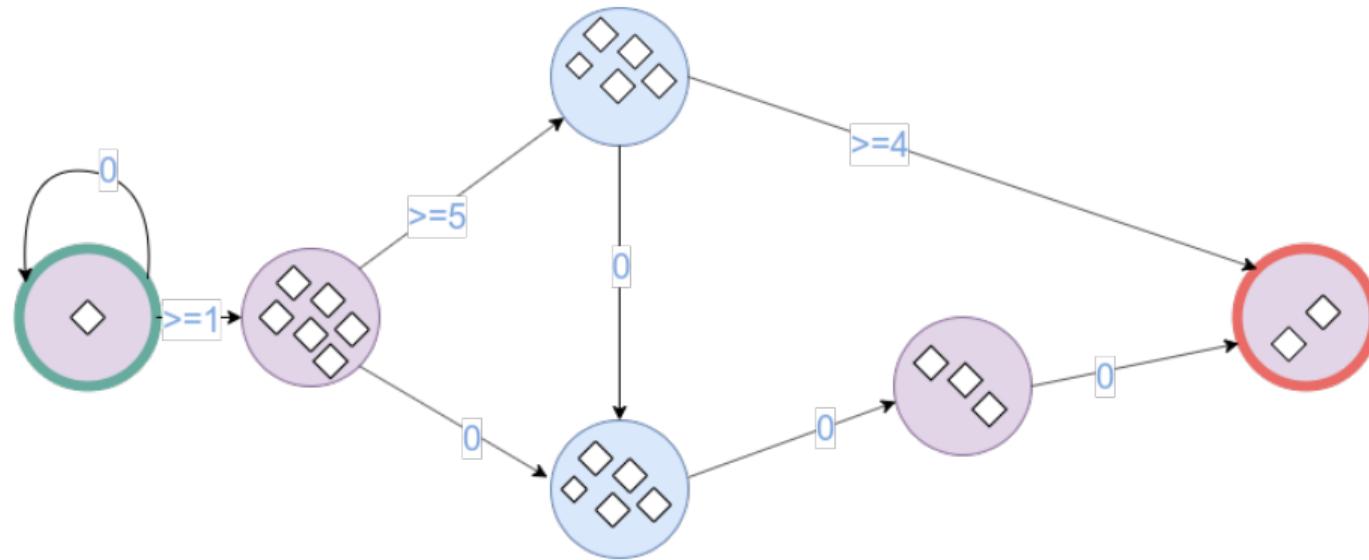
4.3 Item-Ebene

## 4.4 Test-Ebene

4.4.1 Graph eines Tests

4.4.2 Knoten-Typen





# Verschiedene Knoten-Typen

## Linear



Muss korrekt sein

Zufällig



Zeit

Items

Punkte

Abbruchkriterium, Einstellung

## Sprung



Muss korrekt sein

Zufällig



Zeit

Items

Punkte

## Adaptiv

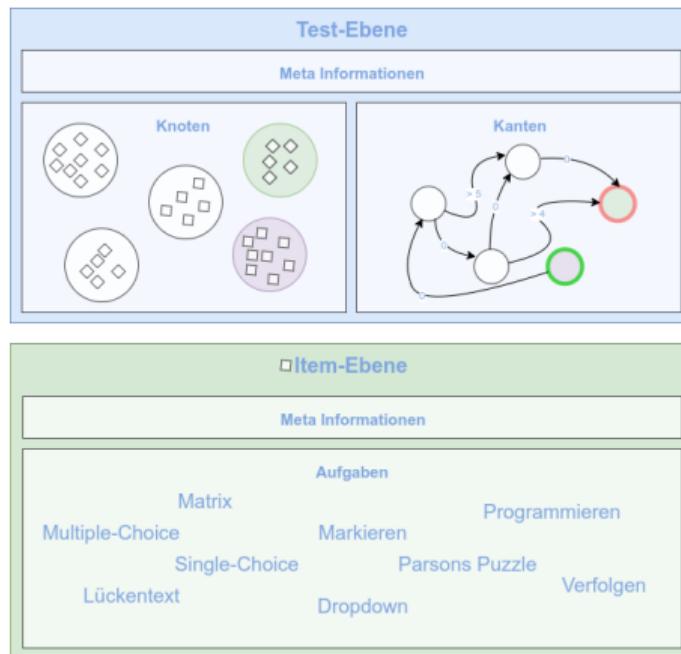


Punkte

Präzision

# Zusammenfassung

# Zusammenfassung



The screenshot shows a digital assessment interface with the following elements:

- Header:** Shows progress (749 / 172), navigation icons, and buttons for "SET\_HIGHLIGHTS" and "DELETE\_HIGHLIGHT".
- Section:** "Aufgabe" (Task).
- Text:** "Klassenbezeichner" (Classifiers). Instructions: "Als Klassenbezeichner werden alle Worte bezeichnet, die auf eine Java-Klasse verweisen. Bitte markieren Sie im folgenden Quelltextfragment alle auftretenden Klassenbezeichner, auch wenn sie doppelt vorkommen." (As class names, all words referring to a Java class are designated. Please mark all occurrences in the following source code fragment, even if they appear twice.)
- Code Editor:** Displays a snippet of Java code with several words highlighted in yellow, indicating they are being marked as class names.
- Buttons:** "ALLE MARKIERUNGEN LÖSCHEN" (Delete all markings).
- Table:** "COMMENTS", "TRANSCRIPT", and "SOLUTION" columns. The "COMMENTS" column contains two entries from "admin" (about 1 hour ago) stating "Really - Delete". The "TRANSCRIPT" column shows a timestamp of 22.7.8 08.8.8. The "SOLUTION" column shows a timestamp of 04.1.9 07.8.
- Footer:** Shows the user's email address: {mike.barkmin, matthias.kramer}@uni-due.de and the text "Automatisches Online-Assessment".

# Ausblick

# Weitere Entwicklungen

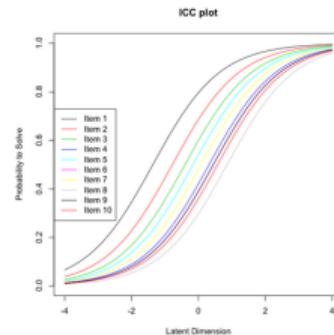
## Verbesserung des Datenschutzes

- Verschlüsselter Knoten-Typen
- Assessment herauslösen



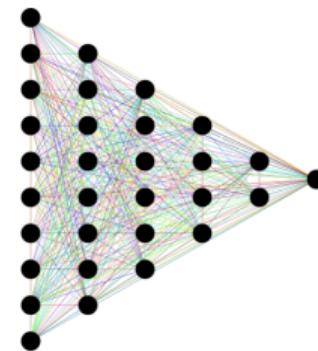
**Abbildung:** Bild von Pete Linforth unter Public Domain via Pixabay License via Pixabay

## Adaptivität Neuer Knoten-Typ



**Abbildung:** Bild von Produnis unter Public Domain via Wikimedia

## Mustererkennung Automatische Auswertung des Bearbeitungsprozesses



**Abbildung:** Bild von GDJ unter Pixabay License via Pixabay

## Falls Sie das Werkzeug einsetzen/mitentwickeln möchten

- Sie sind zur Mitarbeit eingeladen!
- 🏠: Öffentliches Gitlab-Repository  
<https://gitlab.com/openpatch>
- Verwendete Technologien
  - Backend: Python, Flask, MariaDB
  - Frontend: JavaScript, React



# Vielen Dank für die Aufmerksamkeit!

## Kontakt

Mike Barkmin & Matthias Kramer

Didaktik der Informatik

Universität Duisburg-Essen

Schützenbahn 70, 45127 Essen

{mike.barkmin, matthias.kramer}@uni-due.de

<http://ude.de/{mba, mk}>

- Barkmin, Mike u. a. (2017). „Code Structure Difficulty in OOP: An Exploration Study Regarding Basic Cognitive Processes“. In: *Proceedings of the 17th Koli Calling Conference on Computing Education Research*. Koli Calling '17. New York, NY, USA: ACM, S. 185–186. ISBN: 978-1-4503-5301-4.
- Corsi, P. M. (1972). *Human Memory and the Medial Temporal Region of the Brain*. McGill Theses.
- Gluga, Richard u. a. (12. Dez. 2011). „An Architecture for Systematic Tracking of Skill and Competence Level Progression in Computer Science“. In: *2nd Annual International Conferences on Computer Science Education: Innovation and Technology (CSEIT 2011)*. Annual International Conferences on Computer Science Education: Innovation and Technology. Global Science & Technology Forum (GSTF).
- Goedicke, Michael und Michael Striewe (2017). „10 Jahre automatische Bewertung von Programmieraufgaben mit JACK - Rückblick und Ausblick“. In: *Lecture Notes in Informatics*. INFORMATIK 2017. Gesellschaft für Informatik, Bonn. ISBN: 978-3-88579-669-5.

## Literaturverzeichnis II

- Helminen, Juha u. a. (2012). „How Do Students Solve Parsons Programming Problems?: An Analysis of Interaction Traces“. In: *Proceedings of the Ninth Annual International Conference on International Computing Education Research. ICER '12.* New York, NY, USA: ACM, S. 119–126. ISBN: 978-1-4503-1604-0.
- Hsiao, I-Han u. a. (1. Jan. 2008). „Web-Based Parameterized Questions for Object-Oriented Programming“. In:
- Joy, Mike u. a. (Sep. 2005). „The Boss Online Submission and Assessment System“. In: *Journal on Educational Resources in Computing* 5.3.
- Kramer, Matthias u. a. (2018). „Automatic Assessment of Source Code Highlighting Tasks: Investigation of Different Means of Measurement“. In: *Proceedings of the 18th Koli Calling International Conference on Computing Education Research. Koli Calling '18.* 00000. New York, NY, USA: ACM, 8:1–8:10. ISBN: 978-1-4503-6535-2.
- Laakso, Mikko u. a. (2004). „Automatic Assessment of Exercises for Algorithms and Data Structures—a Case Study with TRAKLA2“. In: *Proceedings of the 4th Finnish/Baltic Sea Conference on Computer Science Education*, S. 28–36.

- Parsons, Dale und Patricia Haden (2006). „Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses“. In: *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52. ACE '06.* Darlinghurst, Australia, Australia: Australian Computer Society, Inc., S. 157–163. ISBN: 978-1-920682-34-7.
- Rajala, Teemu u. a. (2016). „Automatically Assessed Electronic Exams in Programming Courses“. In: *Proceedings of the Australasian Computer Science Week Multiconference. ACSW '16.* New York, NY, USA: Association for Computing Machinery. ISBN: 978-1-4503-4042-7.
- Striewe, Michael u. a. (2017). „Ein Lückentext-Test Zur Beherrschung Einer Programmiersprache“. In: *Bildungsräume 2017.* Hrsg. von Christoph Igel u. a. Gesellschaft für Informatik, Bonn, S. 261–266.