Automatic Evaluation of Tasks for Instantaneous Diagnostics in Computer Science Lessons

Seminar - USI - Faculty of Informatics

Mike Barkmin ■ 26 February 2020

Outline

1. Introduction
2. Background
3. Considerations
4. The Online-Assessment-System
5. Summary
6. Next Steps

Figure: Picture of TeroVesalainen under Pixabay License via Pixabay

Introduction

Who am I?

Mike Barkmin

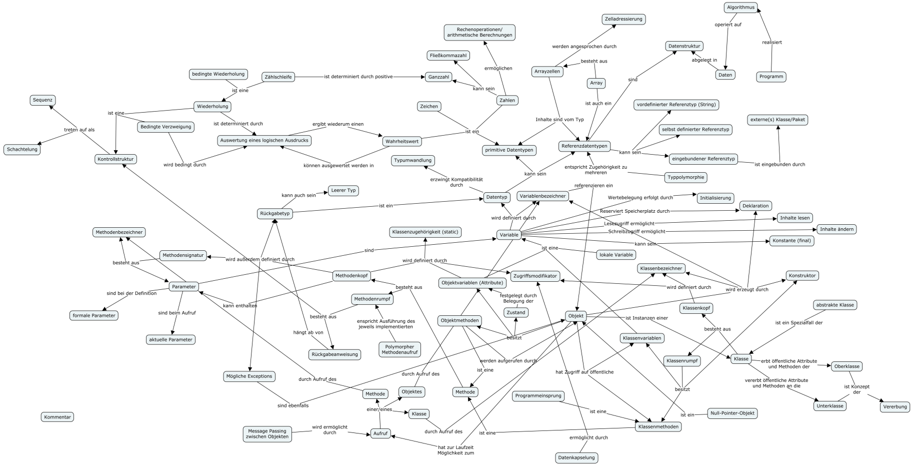
Computer Science Education Research Group

University of Duisburg-Essen, Germany

What is my main research area?

UNIVERSITÄT
DUISBURG
ESSEN

Open-Minded



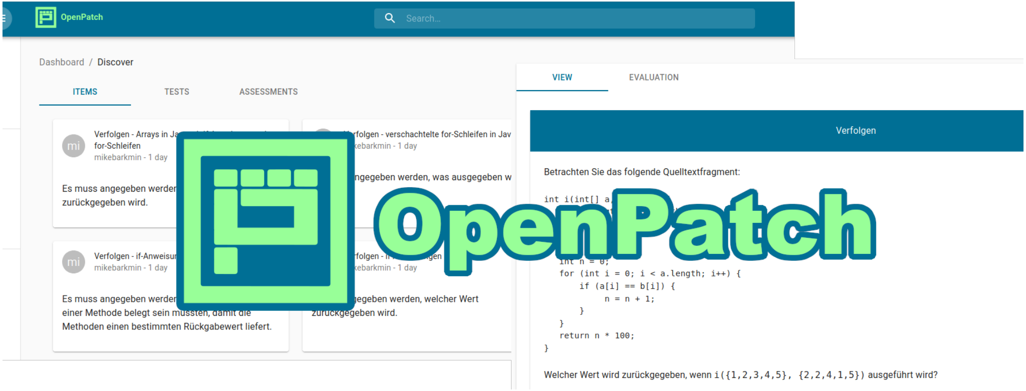
CSE

26 February 2020

mike.barkmin@uni-due.de

Automatic Evaluation and Visualization of Assessments

What will I show you today?



Verfolgen - Arrays in Ja
for-Schleifen
mikebarkmin - 1 day

Es muss angegeben werden
zurückgegeben wird.

rfolgen - verschachtelte for-Schleifen in Jav
kebarkmin - 1 day

ngegeben werden, was ausgegeben w

Verfolgen - if-Anweisu
mikebarkmin - 1 day

Es muss angegeben werden
einer Methode belegt sein mussten, damit die Methoden einen bestimmten Rückgabewert liefert.

folgen - if-
kebarkmin - 1 day

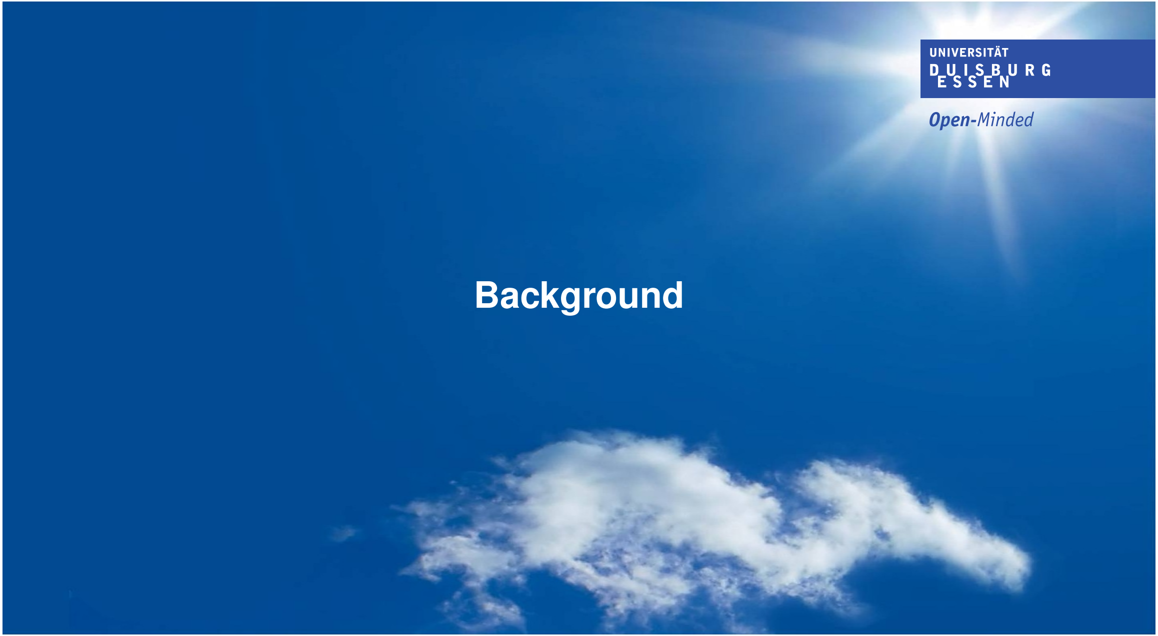
gegeben werden, welcher Wert zurückgegeben wird.

Verfolgen

Betrachten Sie das folgende Quelltextfragment:

```
int i(int[] a,

    int n = 0;
    for (int i = 0; i < a.length; i++) {
        if (a[i] == b[i]) {
            n = n + 1;
        }
    }
    return n * 100;
}
```

Welcher Wert wird zurückgegeben, wenn `i({1,2,3,4,5}, {2,2,4,1,5})` ausgeführt wird?



Background

Background

In studies about the structure of programming knowledge we encountered some problems

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Background

In studies about the structure of programming knowledge we encountered some problems

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage
- Complex task formats are difficult to realize

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage
- Complex task formats are difficult to realize
- Feedback for teachers is staggered

Figure: Picture of Ag Ku under Pixabay License via Pixabay

Considerations

Considerations I

- Webapplication (WA)
- No need for a user account ⇒ Access to the test with a token (NUA)
- Analysis of the problem-solving-capabilities through capturing the interactions (UIT)
- GDPR: partly encrypted submissions and self-hostable (DS)
- Ability to create items and tests (ITE)
- Ability to create new task formats (CE)
- Ability to download all data for further analysis or provided analysis

Considerations II

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

- Additionally, we analyzed other systems (Mooshak, Bottlenose, CourseMarker, WeBWorK and more) as well, but none fitted our needs
- ⇒ Custom development was necessary

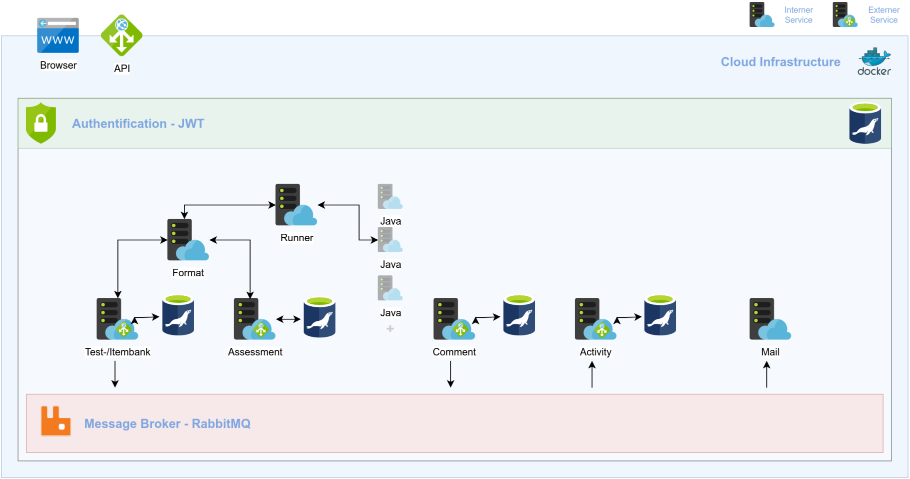
The Online-Assessment-System

The Online-Assessment-System

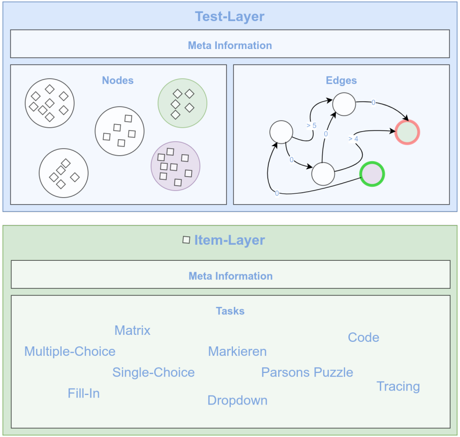
1. Introduction
2. Background
3. Considerations
4. The Online-Assessment-System
    - 4.1 Technical Realization
    - 4.2 Conceptual Realization
    - 4.3 Item-Layer
5. Summary
6. Next Steps

Figure: Picture of TeroVesalainen under Pixabay License via Pixabay

Technical Realization



Conceptual Realization



Test-Layer

Meta Information

Nodes

Edges

□ Item-Layer

Meta Information

Tasks

Matrix

Code

Multiple-Choice

Markieren

Single-Choice

Parsons Puzzle

Tracing

Fill-In

Dropdown

Item-Layer

4. The Online-Assessment-System

4.1 Technical Realization

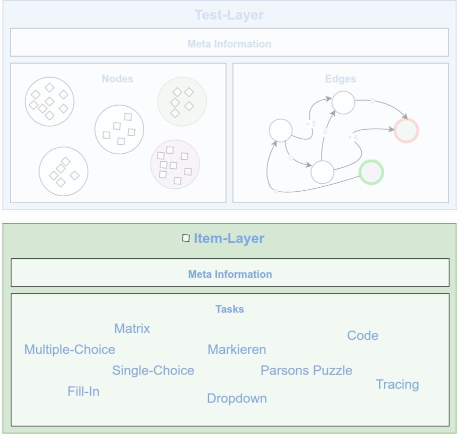
4.2 Conceptual Realization

4.3 Item-Layer

4.3.1 Analog to Digital

4.3.2 Authentic Task Formats

4.3.3 Examination of the Process



Test-Layer

Meta Information

Nodes

Edges

□ Item-Layer

Meta Information

Tasks

Matrix

Code

Multiple-Choice

Markieren

Single-Choice

Parsons Puzzle

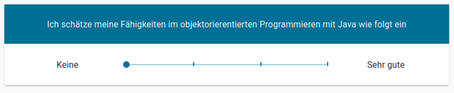
Tracing

Fill-In

Dropdown

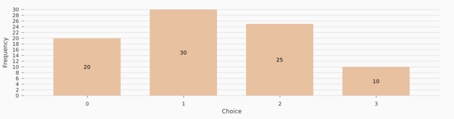
Analog to Digital I

- Scale
  - Digitize analog task formats
  - Makes faster evaluation possible
  - Instantaneous visualization
  - **Evaluation**: Choice
  - **Diagnostic Visualization**: Barchart



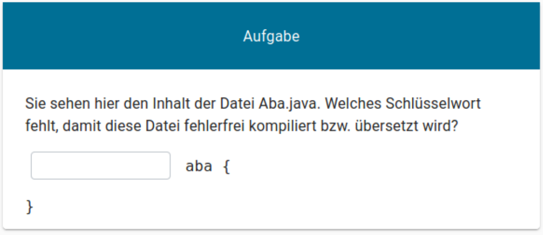
Ich schätze meine Fähigkeiten im objektorientierten Programmieren mit Java wie folgt ein

Keine ● Sehr gute



Analog to Digital II

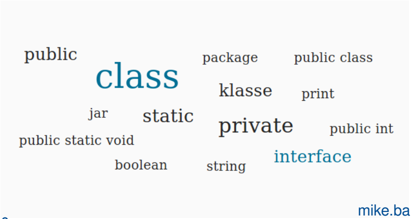
- Fill-in
  - No “handwriting recognition”
  - **Evaluation**: Regular expressions e.g.: “[Ii]nterface|[Cc]lass”
  - **Diagnostic Visualization**: Word-cloud
  - For use in an empirical study see Striewe et al. (2017)



Aufgabe

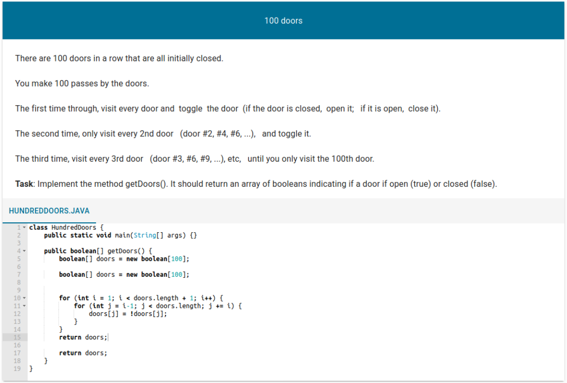
Sie sehen hier den Inhalt der Datei Aba.java. Welches Schlüsselwort fehlt, damit diese Datei fehlerfrei kompiliert bzw. übersetzt wird?

```
[          ] aba {

}
```



Authentic Task Formats

- Desirable to use more authentic task formats
- We implemented a source code runner for this
- Source code will be compiled and tested on our servers
- **Evaluation**: Unittests
- **Diagnostic Visualization**: Currently Missing (Percentage of correct unittests, average time for execution)



100 doors

There are 100 doors in a row that are all initially closed.

You make 100 passes by the doors.

The first time through, visit every door and toggle the door (if the door is closed, open it; if it is open, close it).

The second time, only visit every 2nd door (door #2, #4, #6, ...), and toggle it.

The third time, visit every 3rd door (door #3, #6, #9, ...), etc, until you only visit the 100th door.

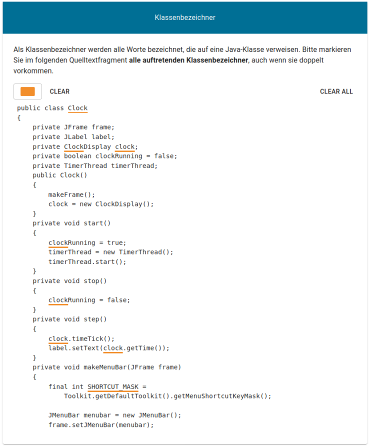
Task: Implement the method getDoors(). It should return an array of booleans indicating if a door if open (true) or closed (false).

HUNDREDDOORS.JAVA

```java
class HundredDoors {
    public static void main(String[] args) {}

    public boolean[] getDoors() {
        boolean[] doors = new boolean[100];

        boolean[] doors = new boolean[100];


        for (int i = 1; i < doors.length + 1; i++) {
            for (int j = i-1; j < doors.length; j += i) {
                doors[j] = !doors[j];
            }
        }
        return doors;

        return doors;
    }
}
```

Development of Complex Task Formats: Highlighting I

- Comparatively simple task format, but authentic
- Was already used by Hauswirth and Adamoli (2013)
- Connects conceptual knowledge with representation of the concepts in a formal language
- Idea: Highlight all spots of `<Concept>` in the given source code

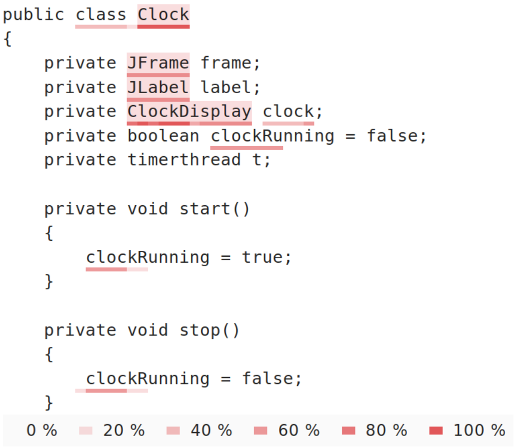


Als Klassenbezeichner werden alle Worte bezeichnet, die auf eine Java-Klasse verweisen. Bitte markieren Sie im folgenden Quelltextfragment **alle auftretenden Klassenbezeichner**, auch wenn sie doppelt vorkommen.

```
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private TimerThread timerThread;
    public Clock()
    {
        makeFrame();
        clock = new ClockDisplay();
    }
    private void start()
    {
        clockRunning = true;
        timerThread = new TimerThread();
        timerThread.start();
    }
    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 = new JMenuBar();
        frame.setJMenuBar(menubar);
```

Development of Complex Task Formats: Highlighting II

- **Evaluation**: Calculate Cohens Kappa and compare to a cutoff score
- **Diagnostic Visualization**: Heatmap
- Evaluation method described in Kramer, Barkmin, Brinda, and Tobinski (2018)
- For use in an empirical study see Kramer, Barkmin, and Brinda (2019)



```
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;
    }
```

Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process

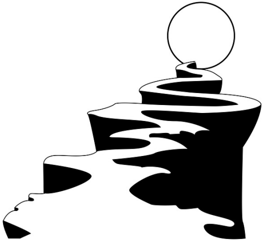


Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay

Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process
- **Solution**: Videorecording of the process
  - 8 students approx. 4h ~ 140GB
  - Manual tagging of events

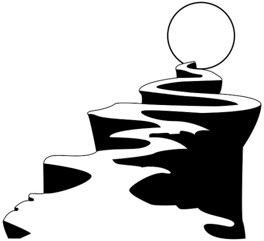


Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay

Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process
- **Solution**: Videorecording of the process
  - 8 students approx. 4h ~ 140GB
  - Manual tagging of events
- **Alternative solution**: Recording of the interactions with the Online-Assessment-System
  - approx. 500 students ~ 20MB
  - Auto tagging of events

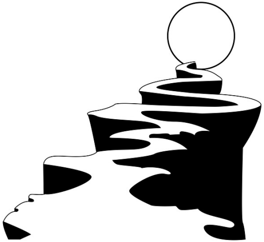


Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay

Recording of the Process

- **Action**: Is dispatched by the user
- **Reducer**: Constructs a new state based on a dispatched action
- **Store**: Contains the current state
- **UI**: Will be rendered depending on the current state in the store

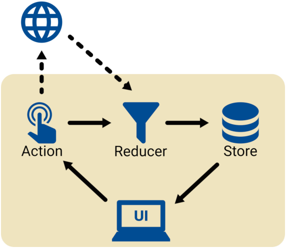
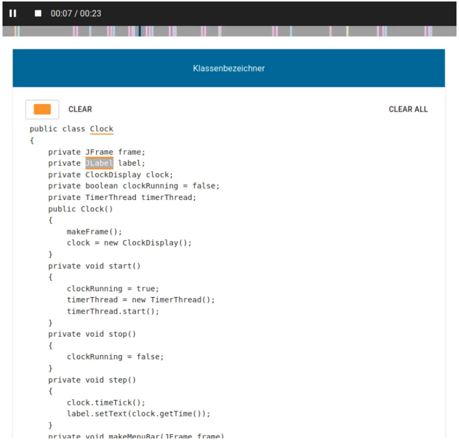
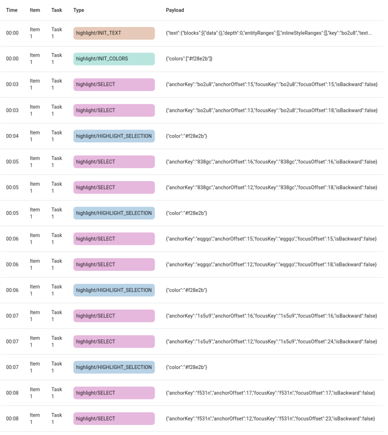


Figure: Action-Reducer-Store see https://redux.js.org

Recording of the Process



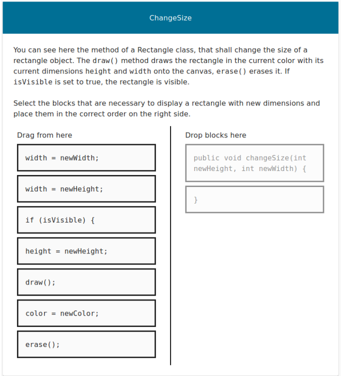
```
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private TimerThread timerThread;
    public Clock()
    {
        makeFrame();
        clock = new ClockDisplay();
    }
    private void start()
    {
        clockRunning = true;
        timerThread = new TimerThread();
        timerThread.start();
    }
    private void stop()
    {
        clockRunning = false;
    }
    private void step()
    {
        clock.timeTick();
        label.setText(clock.getTime());
    }
    private void makeMenuBar(JFrame frame)
```



Time	Item	Task	Type	Payload
00:00	Item 1	Task 1	highlight/INIT_TEXT	{"text":{"blocks":[{"data":{},"depth":0,"entityRanges":[],"inlineStyleRanges":[],"key":"bo2u8","text...
00:00	Item 1	Task 1	highlight/INIT_COLORS	{"colors":["#f28e2b"]}
00:03	Item 1	Task 1	highlight/SELECT	{"anchorKey":"bo2u8","anchorOffset":15,"focusKey":"bo2u8","focusOffset":15,"isBackward":false}
00:03	Item 1	Task 1	highlight/SELECT	{"anchorKey":"bo2u8","anchorOffset":13,"focusKey":"bo2u8","focusOffset":18,"isBackward":false}
00:04	Item 1	Task 1	highlight/HIGHLIGHT_SELECTION	{"color":"#f28e2b"}
00:05	Item 1	Task 1	highlight/SELECT	{"anchorKey":"838gc","anchorOffset":16,"focusKey":"838gc","focusOffset":16,"isBackward":false}
00:05	Item 1	Task 1	highlight/SELECT	{"anchorKey":"838gc","anchorOffset":12,"focusKey":"838gc","focusOffset":18,"isBackward":false}
00:05	Item 1	Task 1	highlight/HIGHLIGHT_SELECTION	{"color":"#f28e2b"}
00:06	Item 1	Task 1	highlight/SELECT	{"anchorKey":"eqgqo","anchorOffset":15,"focusKey":"eqgqo","focusOffset":15,"isBackward":false}
00:06	Item 1	Task 1	highlight/SELECT	{"anchorKey":"eqgqo","anchorOffset":12,"focusKey":"eqgqo","focusOffset":18,"isBackward":false}
00:06	Item 1	Task 1	highlight/HIGHLIGHT_SELECTION	{"color":"#f28e2b"}
00:07	Item 1	Task 1	highlight/SELECT	{"anchorKey":"1s5u9","anchorOffset":16,"focusKey":"1s5u9","focusOffset":16,"isBackward":false}
00:07	Item 1	Task 1	highlight/SELECT	{"anchorKey":"1s5u9","anchorOffset":12,"focusKey":"1s5u9","focusOffset":24,"isBackward":false}
00:07	Item 1	Task 1	highlight/HIGHLIGHT_SELECTION	{"color":"#f28e2b"}
00:08	Item 1	Task 1	highlight/SELECT	{"anchorKey":"f531n","anchorOffset":17,"focusKey":"f531n","focusOffset":17,"isBackward":false}
00:08	Item 1	Task 1	highlight/SELECT	{"anchorKey":"f531n","anchorOffset":12,"focusKey":"f531n","focusOffset":23,"isBackward":false}

Making use of the Recordings

Making use of the Recordings - Visualization



ChangeSize

You can see here the method of a Rectangle class, that shall change the size of a rectangle object. The draw() method draws the rectangle in the current color with its current dimensions height and width onto the canvas, erase() erases it. If isVisible is set to true, the rectangle is visible.

Select the blocks that are necessary to display a rectangle with new dimensions and place them in the correct order on the right side.

Drag from here

```
width = newWidth;
```

```
width = newHeight;
```

```
if (isVisible) {
```

```
height = newHeight;
```

```
draw();
```

```
color = newColor;
```

```
erase();
```

Drop blocks here

```
public void changeSize(int
newHeight, int newWidth) {
```

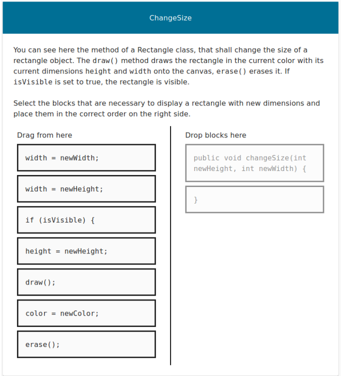
```
}
```

Initial State (left: source, right: user)

- Based on the idea of Parsons and Haden (2006)
- Our actions for Parson Puzzles
  - MOVE_FROM_SOURCE_TO_USER (sourceId, userId)
  - MOVE_FROM_USER_TO_SOURCE (userId, sourceId)
  - MOVE_WITHIN_USER (userId1, userId2)
- What happens, when the action

  MOVE_FROM_SOURCE_TO_USER (1, 1) is dispatched?

Making use of the Recordings - Visualization



ChangeSize

You can see here the method of a Rectangle class, that shall change the size of a rectangle object. The `draw()` method draws the rectangle in the current color with its current dimensions `height` and `width` onto the canvas, `erase()` erases it. If `isVisible` is set to true, the rectangle is visible.

Select the blocks that are necessary to display a rectangle with new dimensions and place them in the correct order on the right side.

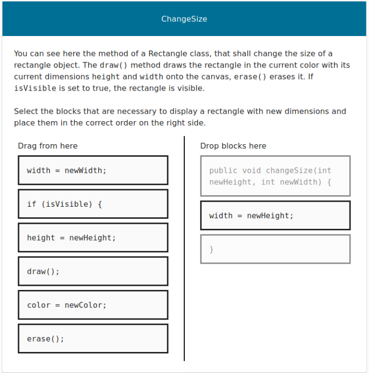
Drag from here

```
width = newWidth;
width = newHeight;
if (isVisible) {
height = newHeight;
draw();
color = newColor;
erase();
```

Drop blocks here

```
public void changeSize(int
newHeight, int newWidth) {
}
```

Initial State (left: source, right: user)



ChangeSize

You can see here the method of a Rectangle class, that shall change the size of a rectangle object. The `draw()` method draws the rectangle in the current color with its current dimensions `height` and `width` onto the canvas, `erase()` erases it. If `isVisible` is set to true, the rectangle is visible.

Select the blocks that are necessary to display a rectangle with new dimensions and place them in the correct order on the right side.

Drag from here

```
width = newWidth;
if (isVisible) {
height = newHeight;
draw();
color = newColor;
erase();
```

Drop blocks here

```
public void changeSize(int
newHeight, int newWidth) {
width = newHeight;
}
```

After dispatching the action MOVE_FROM_SOURE_TO_USER (1, 1)

Making use of the Recordings - Visualization

- Visualization as a directed graph
- Each node represents a state of the parsons puzzle
  - star-shape indicates start state
  - green indicates correct state
- Each edge represents the dispatch of an action
- Number and thickness indicating the frequency

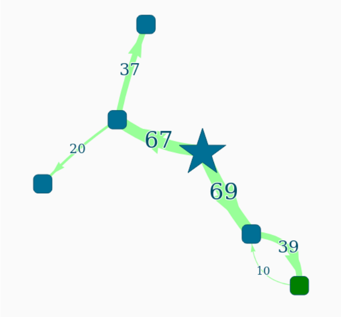
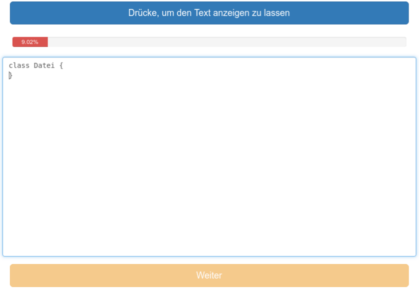
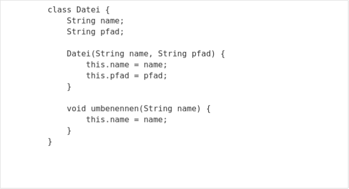


Figure: Visualization of 136 processes based on Helminen et al. (2012)

Making use of the Recordings - Cognitive Structures

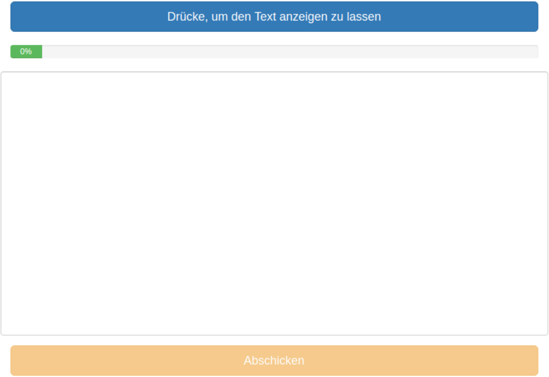


```
class Datei {
}
```



```
class Datei {
    String name;
    String pfad;

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

    void umbenennen(String name) {
        this.name = name;
    }
}
```

Making use of the Recordings - Cognitive Structures



- Actions for task format memorize
  - INSERT_CHAR (charId, pos)
  - REMOVE_CHAR (pos)
  - OPEN_MEMORIZE
  - CLOSE_MEMORIZE

Making use of the Recordings - Cognitive Structures

- Every keystroke is recorded
- Many actions are hard to analyze
- Actions must be combined to reduce complexity
- Memorize-Phases (Blue), Write-Phases (Green) and Pause-Phases (Lightblue)
- Empirical study see Barkmin et al. (2017)

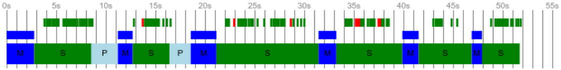
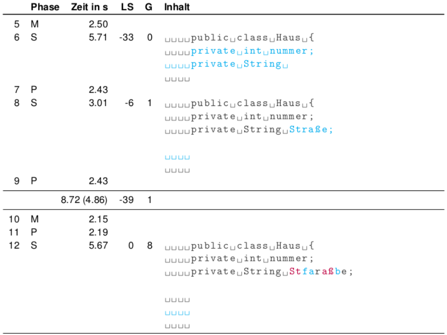


Figure: Timeline of one process



	Phase	Zeit in s	LS	G	Inhalt
5	M	2.50			

6	S	5.71	-33	0	␣␣␣␣public␣class␣Haus␣{
					␣␣␣␣private␣int␣nummer;
					␣␣␣␣private␣String␣
					␣␣␣␣

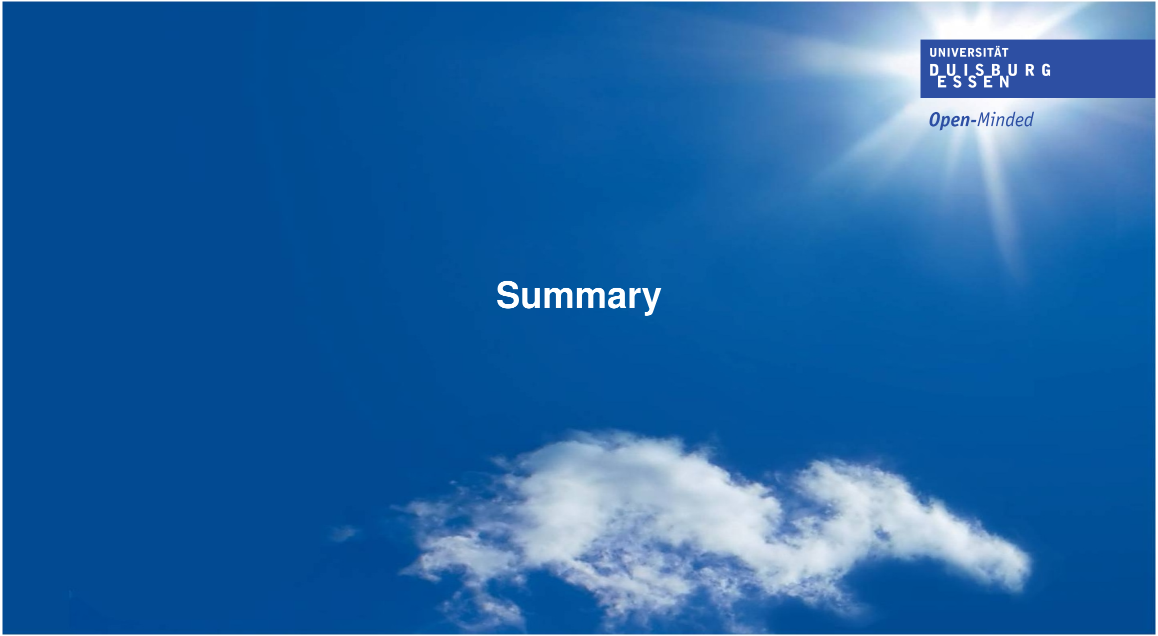
7	P	2.43			

8	S	3.01	-6	1	␣␣␣␣public␣class␣Haus␣{
					␣␣␣␣private␣int␣nummer;
					␣␣␣␣private␣String␣Straße;
					
					␣␣␣␣
					␣␣␣␣

9	P	2.43			
		8.72 (4.86)	-39	1	
10	M	2.15			
11	P	2.19			

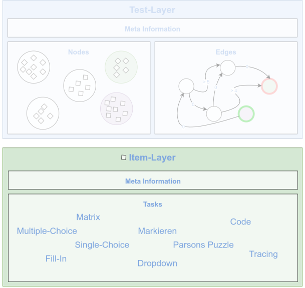
12	S	5.67	0	8	␣␣␣␣public␣class␣Haus␣{
					␣␣␣␣private␣int␣nummer;
					␣␣␣␣private␣String␣Stfaraßbe;
					
					␣␣␣␣
					␣␣␣␣
					␣␣␣␣

Figure: Transcript of one process using combined actions



Summary

Summary



Test-Layer

Meta Information

Nodes

Edges

□ Item-Layer

Meta Information

Tasks

Matrix

Code

Multiple-Choice

Markieren

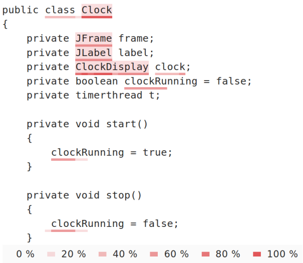
Single-Choice

Parsons Puzzle

Tracing

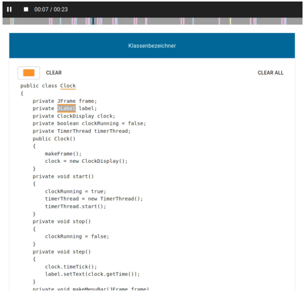
Fill-In

Dropdown

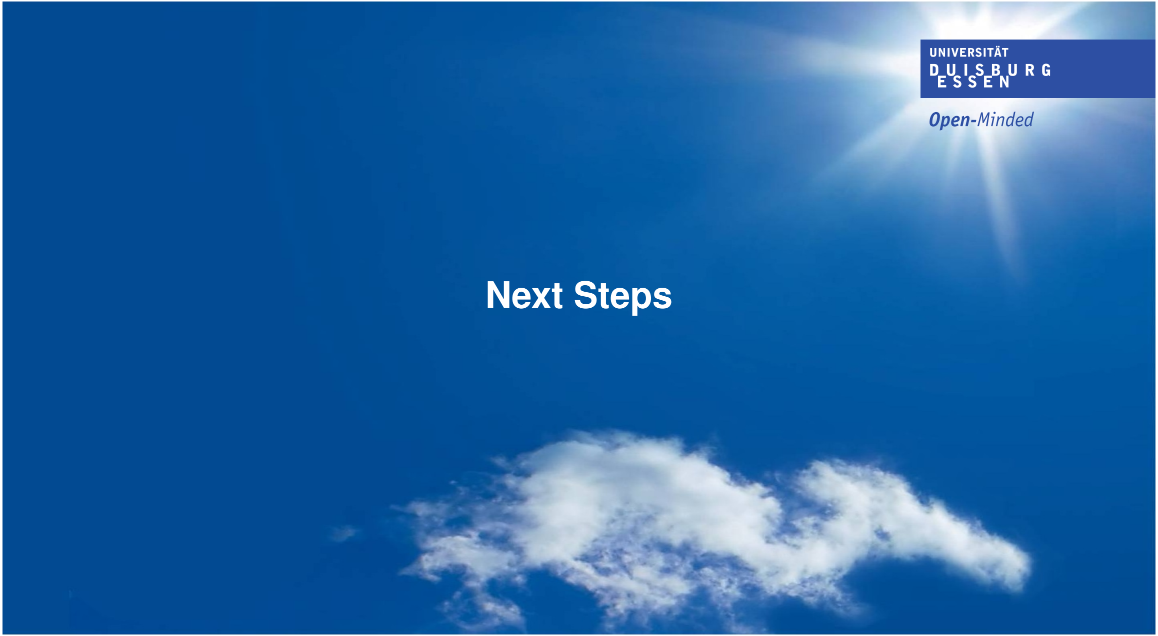


```
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private timerthread t;

    private void start()
    {
        clockRunning = true;
    }

    private void stop()
    {
        clockRunning = false;
    }
```

0 % ▭ 20 % ▭ 40 % ▭ 60 % ▭ 80 % ▭ 100 %



```
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private TimerThread timerThread;
    public Clock()
    {
        makeFrame();
        clock = new ClockDisplay();
    }
    private void start()
    {
        clockRunning = true;
        timerThread = new TimerThread();
        timerThread.start();
    }
    private void stop()
    {
        clockRunning = false;
    }
    private void step()
    {
        clock.timeTick();
        label.setText(clock.getTime());
    }
    private void makeMenuBar(JFrame frame)
```



Next Steps

Next Steps

Teacher - Visualization

Study the handling of the visualizations by teachers

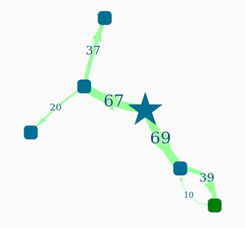


Figure: Picture of JESHOOTS-com under Pixabay License via Pixabay

Pattern-Recognition

Automatic Evaluation of the Process

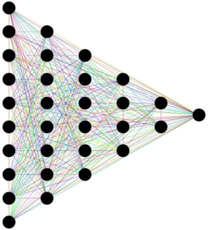


Figure: Picture of GDJ under Pixabay License via Pixabay

Thank you! Any Questions?

Source Code: https://gitlab.com/openpatch

Website: https://openpatch.app

Contact
Mike Barkmin
Computer Science Education Research Group
Universität Duisburg-Essen
Schützenbahn 70, 45127 Essen
mike.barkmin@uni-due.de
http://udue.de/mba

References I

Barkmin, Mike et al. (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, pp. 185–186. ISBN: 978-1-4503-5301-4.

Gluga, Richard et al. (Dec. 12, 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 and 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.

References II

Hauswirth, Matthias and Andrea Adamoli (May 1, 2013). “Teaching Java Programming with the Informa Clicker System”. In: *Science of Computer Programming*. Special Section: Principles and Practice of Programming in Java 2009/2010 & Special Section: Self-Organizing Coordination 78.5, pp. 499–520.

Helminen, Juha et al. (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, pp. 119–126. ISBN: 978-1-4503-1604-0.

Hsiao, I-Han et al. (Jan. 1, 2008). “Web-Based Parameterized Questions for Object-Oriented Programming”. In:

Joy, Mike et al. (Sept. 2005). “The Boss Online Submission and Assessment System”. In: *Journal on Educational Resources in Computing* 5.3.

References III

Kramer, Matthias, Mike Barkmin, and Torsten Brinda (July 2, 2019). “Identifying Predictors for Code Highlighting Skills: A Regressional Analysis of Knowledge, Syntax Abilities and Highlighting Skills”. In: *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. ITiCSE ’19: Innovation and Technology in Computer Science Education. Aberdeen Scotland Uk: ACM, pp. 367–373. ISBN: 978-1-4503-6895-7.

Kramer, Matthias, Mike Barkmin, Torsten Brinda, and David Tobinski (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 et al. (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*, pp. 28–36.

References IV

Parsons, Dale and 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., pp. 157–163. ISBN: 978-1-920682-34-7.

Rajala, Teemu et al. (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 et al. (2017). “Ein Lückentext-Test Zur Beherrschung Einer Programmiersprache”. In: *Bildungsräume 2017*. Ed. by Christoph Igel et al. Gesellschaft für Informatik, Bonn, pp. 261–266.