

Vorlesung Informatikdidaktik

Ein Zugang zum Konzept der Rekursion für den
Informatikunterricht in Schulen

Lehrprobe

Barbara Sabitzer

School of Education – Institut für Informatikdidaktik

Alpen-Adria-Universität Klagenfurt

barbara.sabitzer@aau.at



Vorlesung – Didaktisches Konzept

- Unterrichtsmethoden: Methodenmix
 - Vortrag
 - Flipped Classroom
 - COOL Informatics
 - ...
- Inhalte
 - (fach)didaktische Grundlagen
 - anhand von aufbereiteten Unterrichtsbeispielen
 - zu verschiedenen Informatikkonzepten
 - nach Wahl der Studierenden
 - Anknüpfung an andere Fächer oder Alltagsthemen

Vorlesung Informatikdidaktik

- **Grundlagen**
 - Didaktik & Informatikdidaktik
 - Pädagogische Psychologie
 - Neurobiologische Grundlagen des Lernens
- **Ziele, Inhalte, Methoden** des Informatikunterrichts (ZIM)
 - Ziele
 - Inhalte
 - **Konzepte – Computational Thinking & Problemlösen**
 - Fundamentale Ideen
 - **Rekursion 1 – Konzept**
 - Anwendung(en) – Computing, Standardsoftware
 - Informatik mit (trotz?) Standardsoftware
 - Rekursion 2 – Umsetzung
 - Technologien & Tools – (Instructional) Technology
 - Methoden
- **Informatikunterricht in der Praxis** – Ausgewählte Aspekte
 - Gender & Diversität
 - ... (je nach Interesse der Studierenden)



Rekursion 1

Konzept verstehen

Grundbegriffe kennenlernen

Einführung im Schulunterricht



Rekursion – Stationen

Mögliche Stationen

- Sehen & Begreifen
- Sprache & Kunst
- Bewegung & Aktivität
- Mathematik & Geometrie
- Algorithmen & Programmieren
- Rekursion & Iteration
- Definitionen & Grundlagen

Prinzipien

COOL Informatics

- Discovery
- Cooperation
- Individuality
- Activity

Sehen & Begreifen (1/2)

- Eine Matrjoschka (ru. Матрёшка)

ist eine Puppe,

in der eine Puppe ist,

in der eine Puppe ist,

in der ...

- Ein Objekt ist rekursiv,

- wenn es sich selbst enthält

- oder durch sich selbst definiert ist.

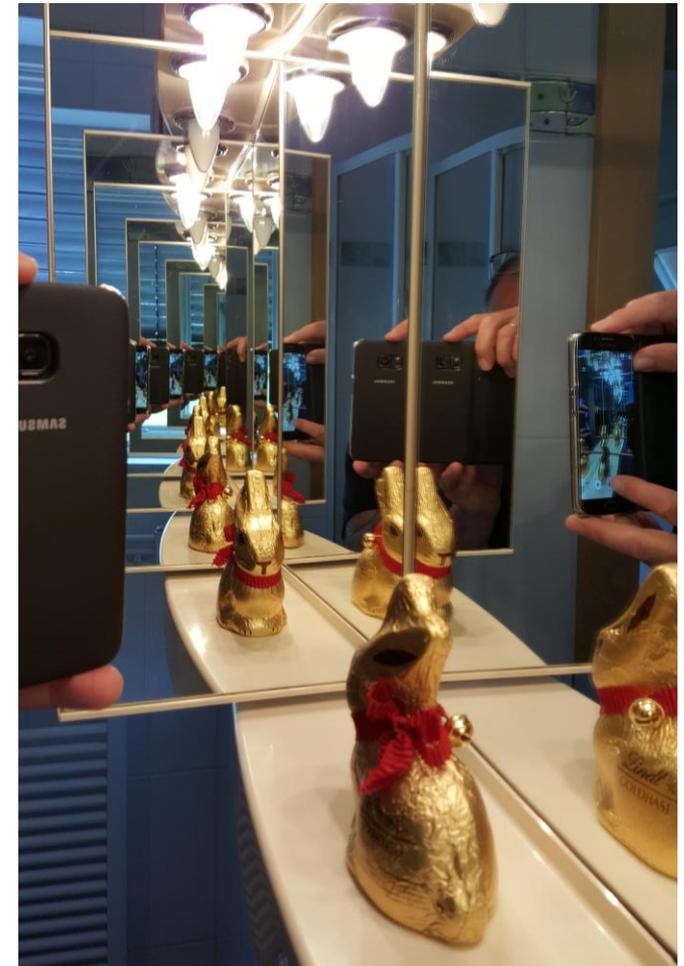


Puppe in der Puppe

Sehen & Begreifen (2/2)



<http://www.lamaisondelavachequirit.com/>



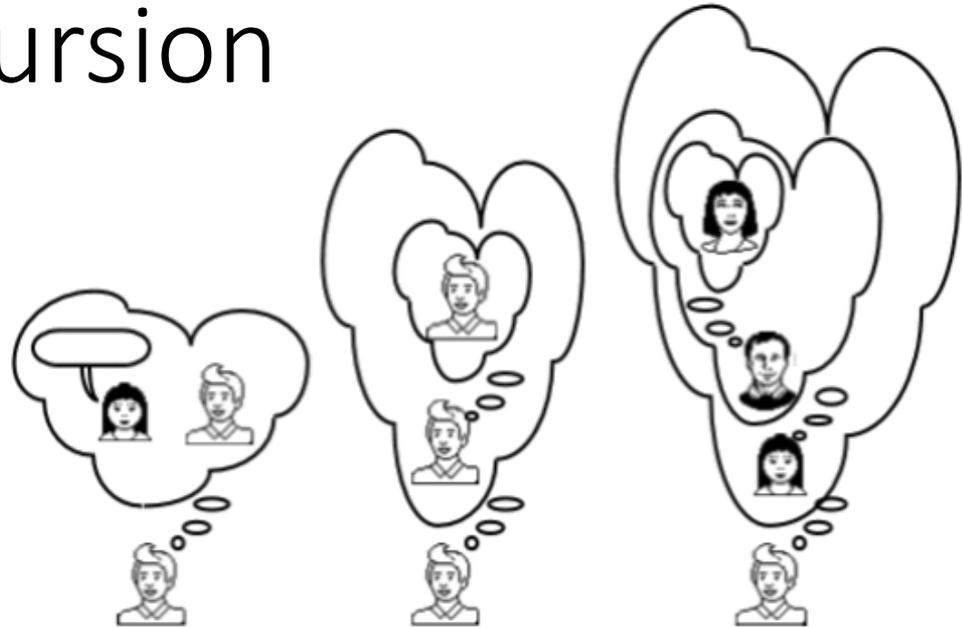
Sprache & unendliche Rekursion

- Ein Lied:
„Ein Hund kam in die Küche und stahl dem Koch ein Ei.
Da nahm der Koch den Löffel und schlug den Hund zu Brei.
Da kamen viele Hunde und gruben ihm ein Grab und
bauten einen Grabstein, auf dem geschrieben stand:
 - „Ein Hund kam in die Küche ...
- Eine Geschichte:
Es war einmal ein Mann, der hatte 7 Kinder, und
die Kinder sprachen: „Vater, erzähl’ uns eine Geschichte“. Da
fang der Vater an:
 - „Es war einmal ein Mann, der hatte 7 Kinder ...

Sprache & Rekursion

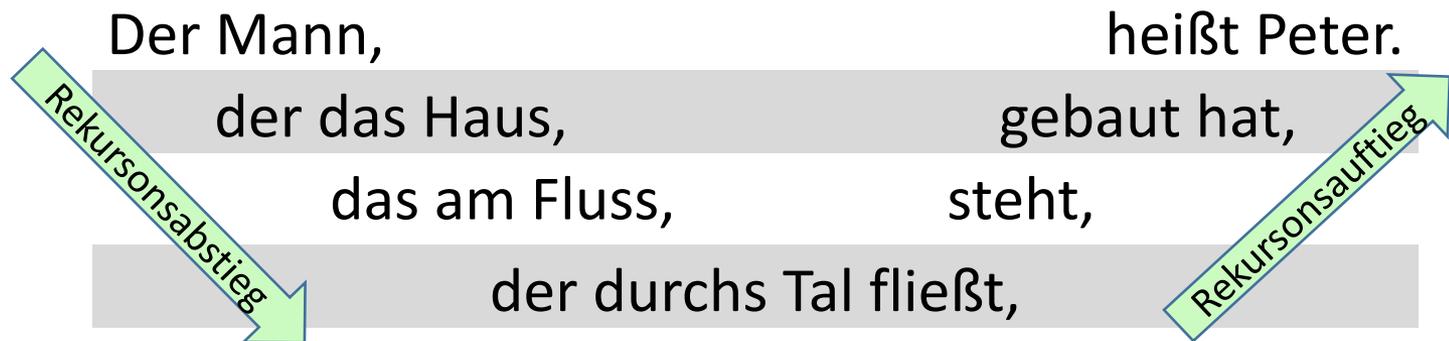
- Rekursives Denken

Max denkt,
dass Karin denkt,
dass Peter...



aus: Schwill, Andreas. "Ab wann kann man mit Kindern Informatik machen."
*INFOS2001-9. GI-Fachtagung Informatik und Schule*GI-Edition (2001): 13-30.

- Verschachtelte Sätze

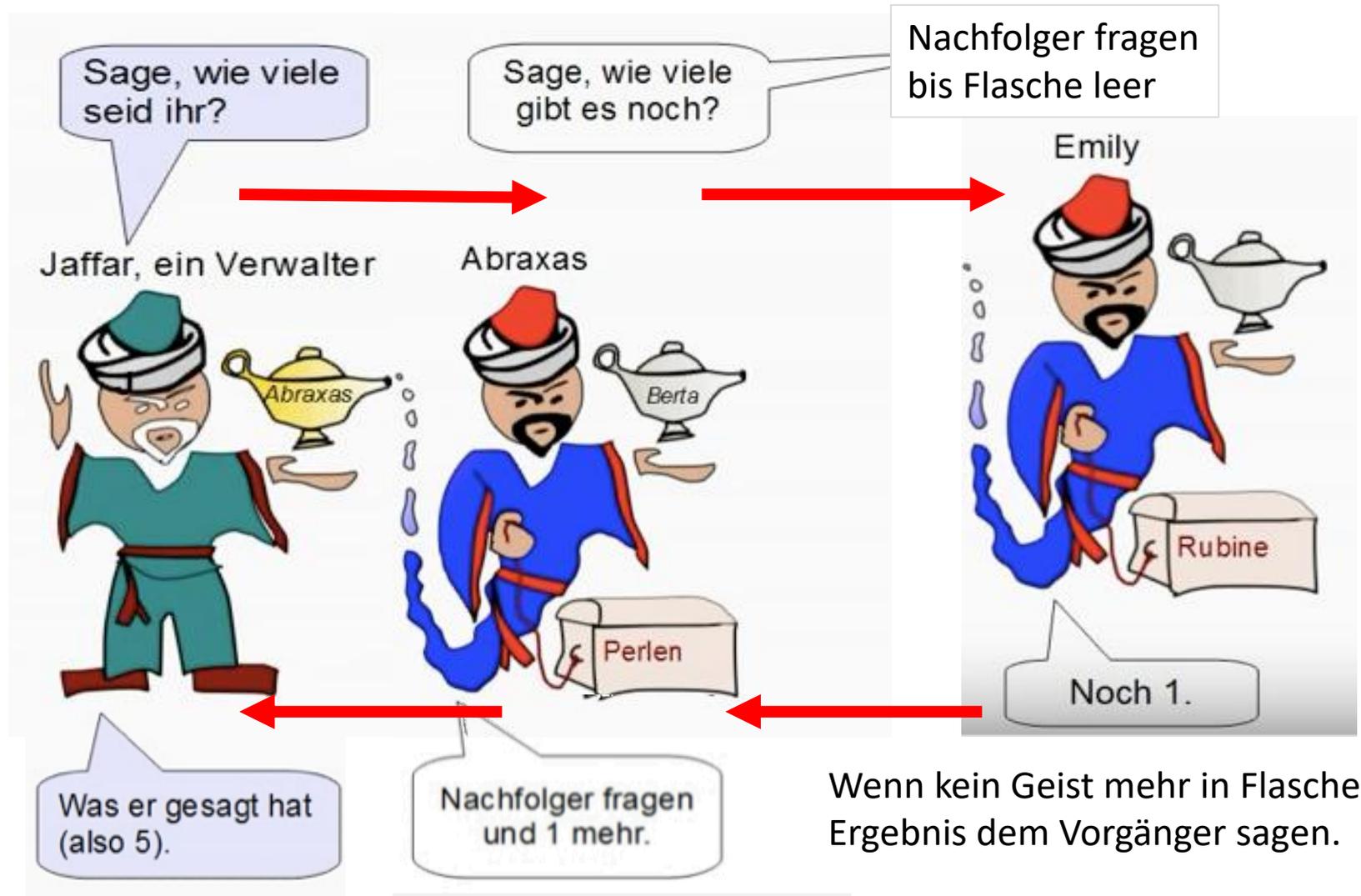


Aufrufstruktur – Kompakte Darstellung sämtlicher rekursiver Aufrufe einer rekursiven Funktion/Methode

Rekursionstiefe – Anzahl der geschachtelten Aufrufe einer rekursiven Funktion/Methode

Rekursion – Verstehen

Jaffar & sein Flaschengeist <https://www.youtube.com/watch?v=0kIPPQD19PM>



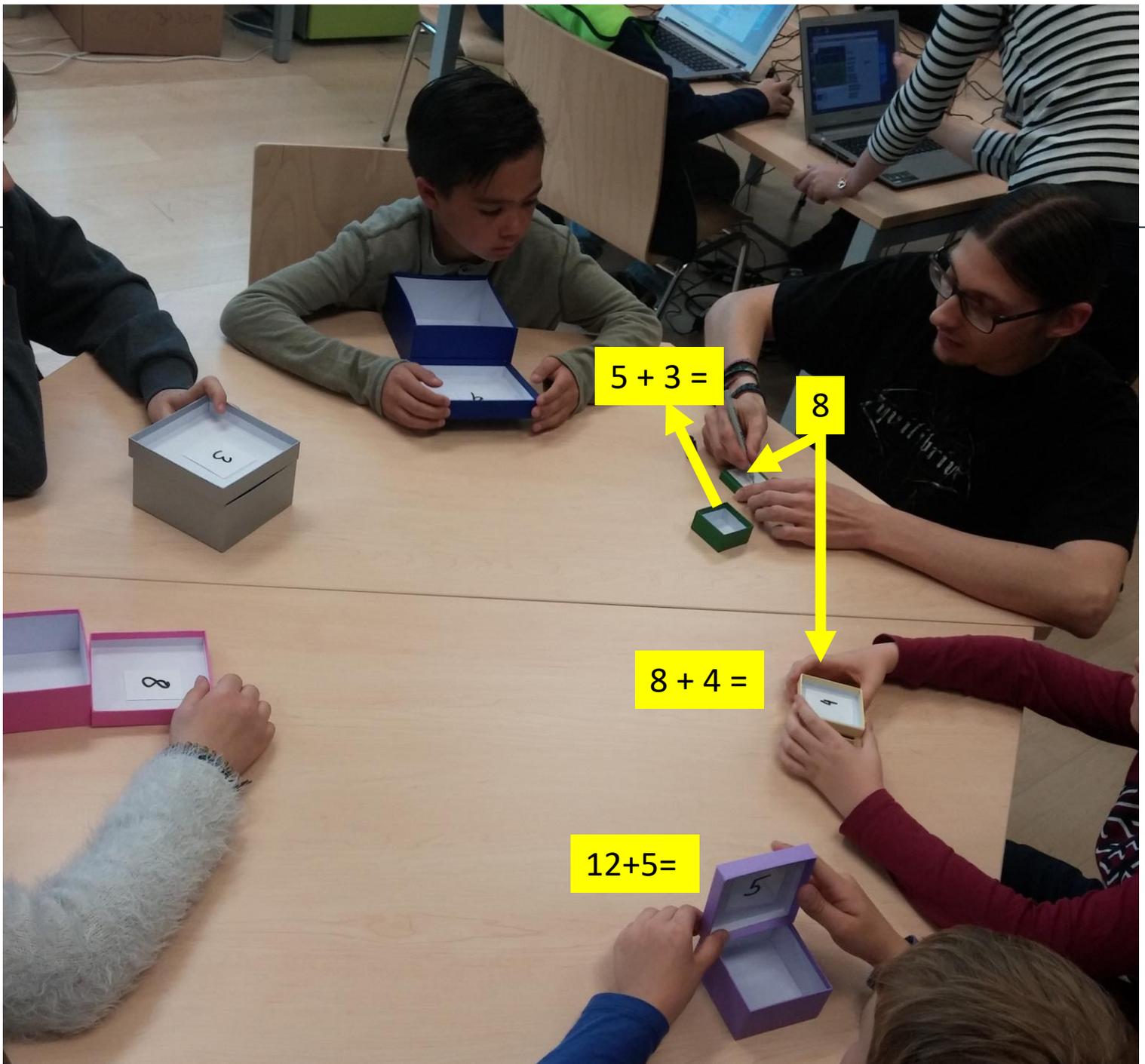
Wenn kein Geist mehr in Flasche:
Ergebnis dem Vorgänger sagen.

Rekursion – Bewegung & Aktivität

- Wie viele gibt es?
 - Flaschengeist nachspielen
 - Matroschkas auspacken
 - Schachteln gesamt



- Schachtelrechnung
 - Rechnung in innerster Schachtel
 - In jedem Deckel
 - Endergebnis ganz außen



Rekursion vs. Iteration

© J. Rau Januar 2013

Vergleich von Rekursion und Iteration

Iteration ist die Wiederholung strukturgleicher Blöcke durch **Aneinanderreihung**



Iteration (Endrekursion):

Man öffnet die größte Puppe, entnimmt die kleinere Puppe und setzt die beiden Teile der größeren Puppe wieder zusammen. Dies wiederholt man, bis die kleinste Puppe ausgepackt ist.



Rekursion ist die Wiederholung strukturgleicher Blöcke durch **Schachtelung**



Echte Rekursion:

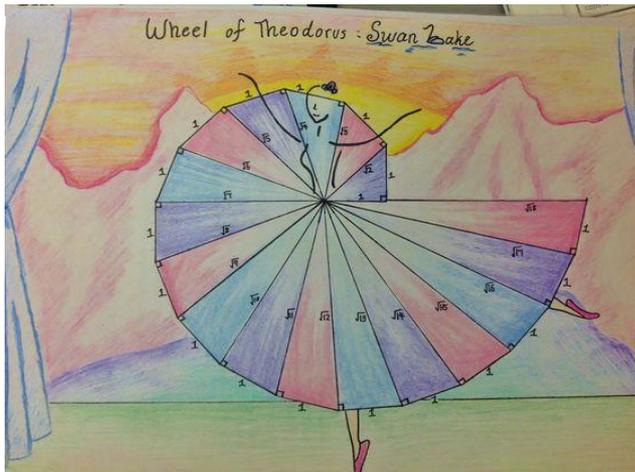
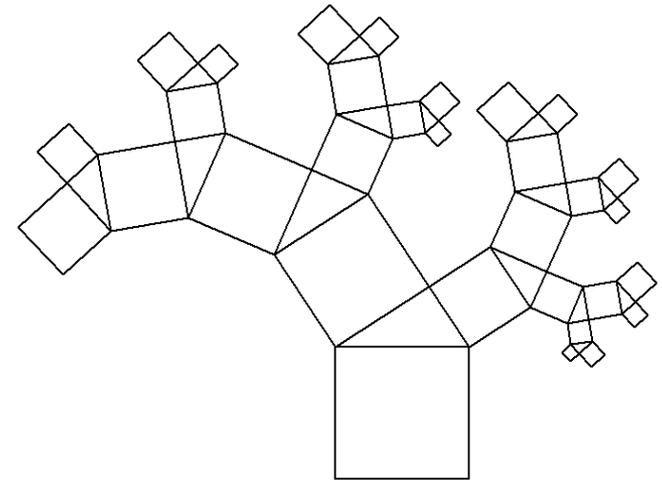
Man öffnet nacheinander jede Puppe, stellt deren 2 Teile geöffnet ab und entnimmt die nächste Puppe. Wenn die kleinste Puppe entnommen wurde, setzt man die jeweiligen zwei Teile der größeren Puppen in umgekehrter Reihenfolge zusammen.



<http://www.j-rau.de/jogohegy/gk12/15RekursiveAlgorithmen.pdf>

Mathematik & Geometrie (1/2)

- Geometrie
 - Der Baum des Pythagoras
 - Das Rad des Theodorus
 - Kochkurve
 - ...



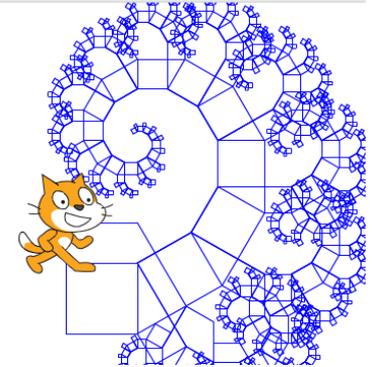
- Mathematik
 - Intelligenztest: Zahlenreihen
 - Fibonacci
 - Fakultät
 - ggT

<https://www.pinterest.com/pin/546694842236347302>

Mathematik & Geometrie (2/2)

```
Wenn  angeklickt  
wische Malspuren weg  
schalte Stift ein  
zeichne Baum 70
```

```
Definiere zeichne Baum länge  
wiederhole 4 mal  
  gehe länge er-Schritt  
  drehe dich  um 90 Grad  
    
  falls länge > 5 dann  
    gehe länge er-Schritt  
    drehe dich  um 30 Grad  
    zeichne Baum länge * 0,5 * Wurzel von 3  
    drehe dich  um 90 Grad  
    gehe länge * 0,5 * Wurzel von 3 er-Schritt  
    zeichne Baum länge / 2  
    gehe länge * -0,5 * Wurzel von 3 er-Schritt  
    drehe dich  um 60 Grad  
    gehe -1 * länge er-Schritt
```



Rekursive Algorithmen (1/4)

Grundstruktur?

Programm Löse (Problem)

Begin

```
//1. Rekursionsbasis im if = Abbruchbedingung
```

```
If Problem einfach
```

```
    Löse Problem direkt
```

```
//2. Rekursionsvorschrift
```

```
Else
```

```
    Zerlege das Problem in kleinere, gleichartige Probleme
```

```
    Löse (kleines Problem)
```

```
    Verwende kleines Problem zur Lösung des Problems
```

```
End
```

End

?



Puppenspielen mit Matrjoschkas??

Programm Auspacken (Matrjoschka)

Begin

```
If kleinste Matrjoschka gefunden  
    Problem gelöst
```

```
Else
```

```
    Öffne die Matrjoschka
```

```
    Entnimm die nächste Matrjoschka
```

```
    Auspacken (Matrjoschka)
```

```
End
```

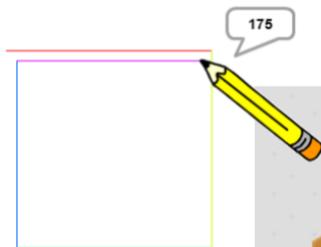
End

?

Algorithmen & Programmieren (2/4)

Rekursion_Spirale

Spirale zeichnen mit Scratch Farben für Rekursionsschritte



Rekursion_Spirale



```
Wenn angeklickt  
wische Malspuren weg  
schalte Stift ein  
Zeichne eine Seite mit Länge 200
```

„normaler“ Aufruf
der Methode

```
Definiere Zeichne eine Seite mit Länge länge  
ändere Stiftfarbe um 40  
falls länge < 0 dann  
sonst  
sage länge für 1 Sek.  
gehe länge er-Schritt  
drehe dich um 90 Grad  
warte 1 Sek.  
Zeichne eine Seite mit Länge länge - 5
```

Abbruchbedingung,
nicht rekursiver
Fall: keine Aktion,
Rückkehr in höhere
Ebene

rekursiver Aufruf
mit vermindertem
Parameter



Algorithmen & Programmieren (3/4)

5.1 Lesecke

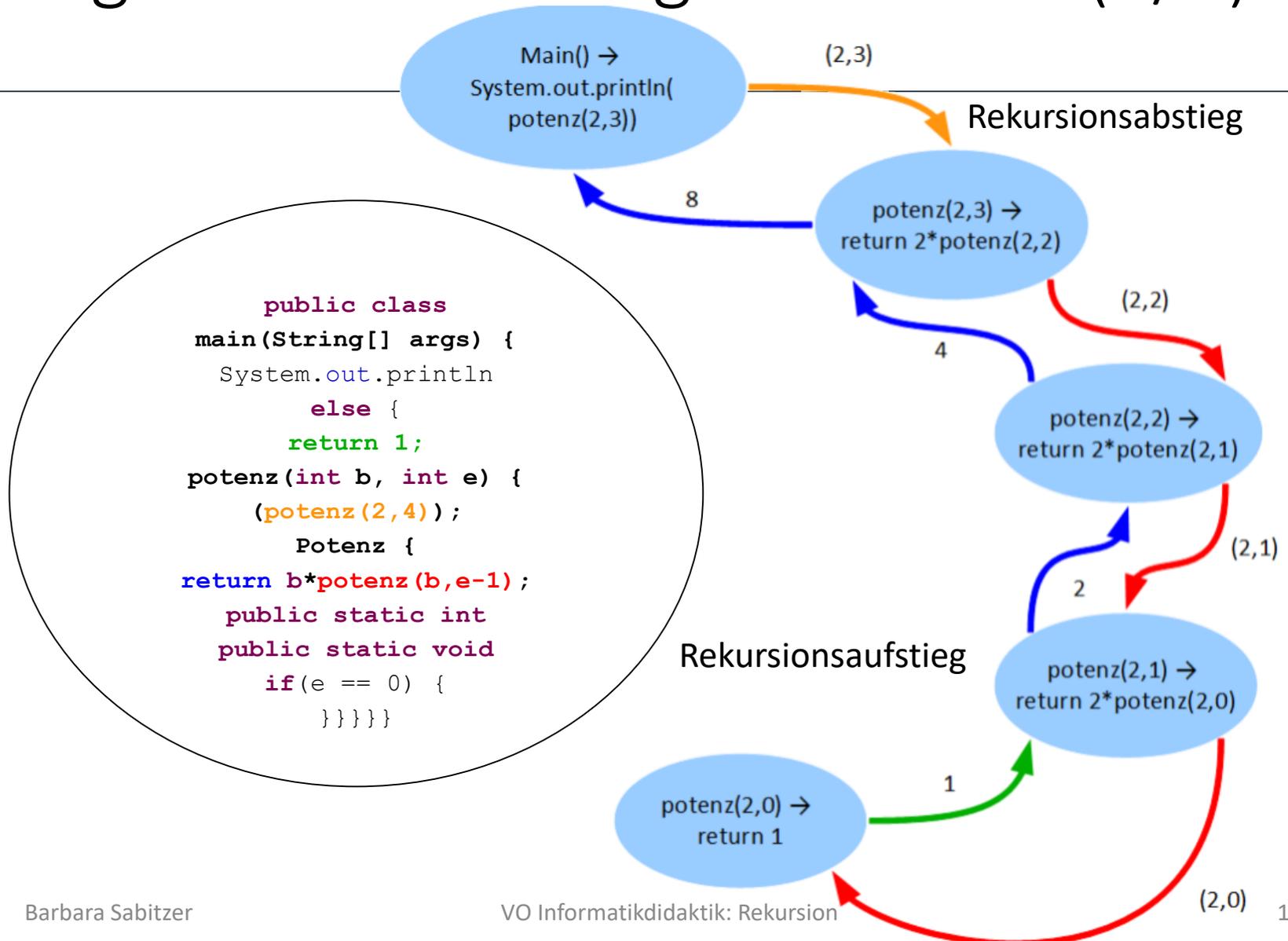
Zeit vergeht

```
public class Endlosrekursion {  
  
    public static void main(String[] args) {  
        long zeit = 0;  
        zeitVergeht(zeit);  
    }  
  
    public static void zeitVergeht(long zeit) {  
        System.out.println(zeit);  
        zeitVergeht(zeit + 1);  
    }  
}
```

5.2 Fragen und Aufgaben zur Lesecke

1. Lesen Sie das obige Beispiel zur Rekursion und probieren Sie es aus.
2. Was passiert? Warum?

Algorithmen & Programmieren (4/4)



Ausgewählte Infos & Links für den Einstieg in die Rekursion

- Leitprogramm Rekursives Programmieren
<http://swisseduc.ch/informatik/programmiersprachen/rekursion/docs/rekursion.pdf>
- <http://www.inf-schule.de/algorithmen/algorithmen/rekursion>
- [Quicksort](#) mit Lego und Stop Motion
- Sortieralgorithmen mit [AlgoRythmics](#)
- Klangsteuerung: [I'm sitting in a room](#)

Klangsteuerung I'm sitting in a room

- I am sitting in a room different from the one you are in now.
 - I am recording the sound of my speaking voice and
 - I am going to play it back into the room again and again
 - until the resonant frequencies of the room reinforce themselves so that any semblance of my speech, with perhaps the exception of rhythm, is destroyed.
 - What you will hear, then, are the natural resonant frequencies of the room articulated by speech.
- I regard this activity not so much as a demonstration of a physical fact, but more as a way to smooth out any irregularities my speech might have.“

Rekursion

Abschluss – Verständnisfragen an die Studierenden

Die Fakultät einer Zahl:

$$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

$$\begin{aligned} 5! &= 5 \cdot 4! \\ &= 5 \cdot (4 \cdot 3!) \\ &= 5 \cdot (4 \cdot (3 \cdot 2!)) \\ &= 5 \cdot (4 \cdot (3 \cdot (2 \cdot 1!))) \\ &= 5 \cdot (4 \cdot (3 \cdot (2 \cdot (1)))) \\ &= 5 \cdot (4 \cdot (3 \cdot (2))) \\ &= 5 \cdot (4 \cdot (6)) \\ &= 5 \cdot (24) \\ &= 120 \end{aligned}$$

Phase des
rekursiven
„**Abstiegs**“
(rekursive Aufrufe)

Phase des
rekursiven
„**Aufstiegs**“
(Rückgabe und
Zusammenbau der
Ergebnisse)