

Softwaretechnik 1 - 5. Tutorium

Tutorium 03

Felix Bachmann | 10.07.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



1 Orga

2 Recap

3 Parallelität

4 Testen

5 Tipps

Nächstes Mal letztes Tutorium

- irgendwelche Wünsche für das letzte Tut?
 - etwas bestimmtes wiederholen?
 - falls euch noch was einfällt, schreibt mir eine Mail
⇒ felix.bachmann@ewetel.net

Nächstes Mal letztes Tutorium

- irgendwelche Wünsche für das letzte Tut?
 - etwas bestimmtes wiederholen?
 - falls euch noch was einfällt, schreibt mir eine Mail
⇒ felix.bachmann@ewetel.net

Evaluation vom letzten Mal

- nochmal Danke fürs Mitmachen!

Nächstes Mal letztes Tutorium

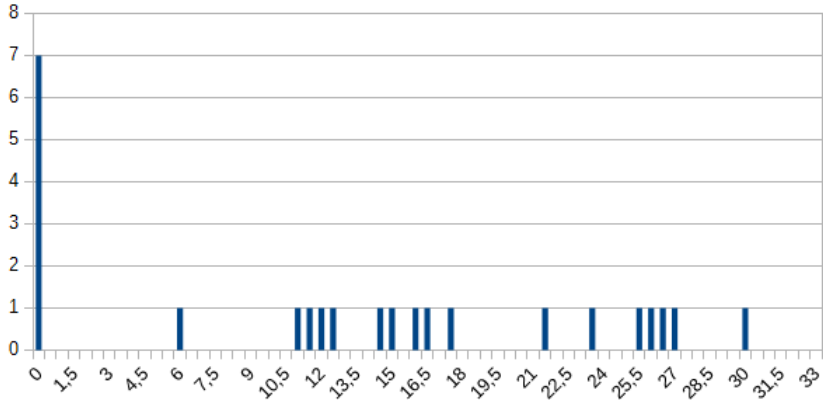
- irgendwelche Wünsche für das letzte Tut?
 - etwas bestimmtes wiederholen?
 - falls euch noch was einfällt, schreibt mir eine Mail
⇒ felix.bachmann@ewetel.net

Evaluation vom letzten Mal

- nochmal Danke fürs Mitmachen!
- häufigster Kritikpunkt: nicht so gut lesbarer Tafelanschrieb
⇒ versuche ich besser zu machen :)

4. Übungsblatt Statistik

n=24



Ø 13 bzw 18,4 von 25+8

4. Übungsblatt - Häufige Fehler (A3)

Aufgabe 3 (GUI für Geometrfy): 6,56 bzw. 11,25 von 10+7

```
// won't work from the jar
File f = new File("src/main/resources/bla.png");

// use one of the following (which one depends on your needs):
this.getClass().getResource("bla.png");
Thread.currentThread().getContextClassLoader().getResource("bla.png");
System.class.getResource("bla.png");
```

4. Übungsblatt - Häufige Fehler (A3)

Aufgabe 3 (GUI für Geometrfy): 6,56 bzw. 11,25 von 10+7

```
// won't work from the jar
File f = new File("src/main/resources/bla.png");

// use one of the following (which one depends on your needs):
this.getClass().getResource("bla.png");
Thread.currentThread().getContextClassLoader().getResource("bla.png");
System.class.getResource("bla.png");
```

- keine leeren JPanels o.a. Objekte benutzen, um Platz zwischen Objekten zu erzeugen
⇒ geht schöner, performanter mit LayoutManagern

4. Übungsblatt - Häufige Fehler (A3)

Aufgabe 3 (GUI für Geometrify): 6,56 bzw. 11,25 von 10+7

```
// won't work from the jar
File f = new File("src/main/resources/bla.png");

// use one of the following (which one depends on your needs):
this.getClass().getResource("bla.png");
Thread.currentThread().getContextClassLoader().getResource("bla.png");
System.class.getResource("bla.png");
```

- keine leeren JPanels o.a. Objekte benutzen, um Platz zwischen Objekten zu erzeugen
⇒ geht schöner, performanter mit LayoutManagern
- `fileChooser.setFileFilter(filter)` anstatt
`fileChooser.addChoosableFileFilter(filter)`

mehr zur Aufgabe 3 nach dem Parallelität-Teil!

5. Übungsblatt Statistik

Allgemein



Aufgabe 1 (Architekturstile): Ø von 5

Aufgabe 1 (Architekturstile): Ø von 5



Aufgabe 2 (Iterator für Plug-Ins): Ø von 6

Aufgabe 2 (Iterator für Plug-Ins): 0 von 6



Aufgabe 2 (Iterator für Plug-Ins): Ø von 6



Aufgabe 3 (Umstrukturierung von Geometrify): Ø von 8



Aufgabe 4 (Reimplementierung von Geometrify): Ø von 7+2

Aufgabe 4 (Reimplementierung von Geometrify): Ø von 7+2



4. Übungsblatt - A3 mit Threads

Aufgabe 3 (GUI für Geometrify): 6,56 bzw. 11,25 von 10+7

```
public Test() {  
    // calling Swing methods from arbitrary threads may result in unexpected behaviour  
    // because most Swing Components are not thread safe!  
    new JFrame("HelloWorld").setVisible(true);  
  
    // instead use the swing event dispatch thread every time you paint, build,... Swing components  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        public void run() {  
            new JFrame("HelloWorld").setVisible(true);  
        }  
    });  
}
```

4. Übungsblatt - A3 mit Threads

Aufgabe 3 (GUI für Geometrify): 6,56 bzw. 11,25 von 10+7

```
// will freeze the gui when the button is clicked
JButton heavy = new JButton("Freeze");
heavy.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // heavy work
    }
});

// will not freeze the gui
JButton light = new JButton("Don't freeze");
light.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // use a new thread to handle heavy work
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() {
                // heavy work
            }
        });
        // starts the thread, "main thread" will return immediately
        t.start();
    }
});
```

Aufgabe 1: Kontrollfluss-orientiertes Testen

- Zwischensprache benutzen
- Definitionen der verschiedenen Abdeckungen anschauen

Aufgabe 1: Kontrollfluss-orientiertes Testen

- Zwischensprache benutzen
- Definitionen der verschiedenen Abdeckungen anschauen

Aufgabe 2: Codeinspektion

- an das Format halten

Aufgabe 3: Parallelisierung von Geomtrify

- Berechnung der Samples parallelisieren
- Zahl der benutzten Threads abhängig machen von der Anzahl der Prozessor-Kerne

Aufgabe 3: Parallelisierung von Geomtrify

- Berechnung der Samples parallelisieren
- Zahl der benutzten Threads abhängig machen von der Anzahl der Prozessor-Kerne

Aufgabe 4: Alternative Parallelisierungsverfahren

- theoretische Überlegungen, was man sonst noch so parallelisieren könnte
- Sinnhaftigkeit, Aufwand, etc. prüfen

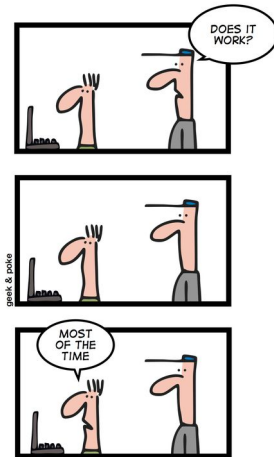
Aufgabe 5: Parallelisierungswettbewerb

- Aufgabe 3 verbessern und Laufzeit messen

Abgabe

- Deadline am 19.7. um 12:00
- Aufgabe 1,2,4 und Beschreibung, Laufzeitprofil von Aufgabe 5 handschriftlich

SIMPLY EXPLAINED



CONCURRENCY