

**JKU**

**JOHANNES KEPLER  
UNIVERSITY LINZ**

# Praktische Informatik 2



Übung, 2017S  
Institut für System Software (SSW)  
DI Eisl & DI Leopoldseder



# Übungsmodus (1)

- Übungsaufgaben
  - **11** Übungszettel zu je **24** Punkte
  - **9 positiv** beurteilte Abgaben nötig
    - ABER: Es werden **11 Übungen gewertet**
    - ACHTUNG: Übung gilt als **abgegeben** wenn **mindestens 6 Punkte** erreicht worden sind
  - Am Ende des Semesters müssen **50%** der Punkte erreicht worden sein  
 $11 * 24 / 2 = 132$
- Beurteilung
  - Abschlusstest am Ende des Semesters: 27.06.2017

# Übungsmodus (2)

- Beurteilung
  - Abschlusstest am Ende des Semesters (24 Punkte)
  - Endnote =  $p_{\text{Test}} * 0.5 + p_{\text{Übung}} * 0.5$
  - **Beide Teile müssen positiv sein**
  - Ab der **2. abgegebenen** Übung wird eine Note ausgestellt
- **LVA Evaluierung im KUSSS ;-)**

# Testablauf

- 45 Minuten
- 24 Punkte
- **Keine Bleistifte oder radierbare Kugelschreiber**
- **FARBEN!!!**
- 4-5 Aufgaben
  - Zeichnen von Algorithmen
  - Code Beispiele
  - Stoff Fragen → Verständnis
- Bei negativem Ergebnis → Nachtest im Herbst

# Stoff der Übung



# Stoff des Semesters

1. Listen
2. Stacks, Queues, Sets
3. Bäume
4. Graphen
5. Hashing
6. Stringsuche
7. Sortieren

# 1 Listen

1. Pointer & Managed Memory (GC)
2. Listen
  - a. Lineare Listen
    - i. Einfach verkettet
    - ii. Doppelt verkettet
    - iii. unsortiert
    - iv. sortiert
    - v. Vorteile / Nachteile / Komplexität / etc.



## 2 Stacks, Queues, Sets

1. Stack
2. Queue

# 3 Bäume

## 1. Binärbäume

### a. Definitionen

- i. parent, root, head, child, sibling, height, width, path, leaf
- ii. Vollständiger Binärbaum

### b. Eigenschaften

### c. Operationen: Einfügen, Löschen, Suchen, Iterieren (Traversierung)

### d. Binäre Suchbäume → Suchbaumeigenschaft

## 2. Balancierte Bäume

### a. 234 Bäume

### b. R/S Bäume

## 3. Heaps

# 4 Graphen

1. Definitionen
  - a. Vertex, Edge, Cycle, Path
  - b. gerichtet & ungerichtet
  - c. gewichtet & ungewichtet
  - d. vollständig (verbunden)
  - e. Speicherdarstellung (Adjazenzliste, AdjazenzMatrix, Direkte Pointer)
2. DFS
3. BFS
4. MST
5. Shortest Path
6. Transitive Hülle - Warshall Algorithmus

# 5 Hashing

1. Schnellste Suchmethode / Speicherung von Key $\leftrightarrow$  Value Paaren
2. Hashfunktionen (32 Bit Integer)  $\rightarrow$  gut Streuung  $\rightarrow$  Primzahlen
3. Kollisionsstrategien
  - a. Linear Probing
  - b. Quadratic Probing
  - c. Separate Chaining
4. Füllgrad, Komplexität, etc.

# 6 Stringsuche

1. Brute Force
2. Boyer Moore

# 7 Sortieren

1. Heapsort
2. Mergesort