

Programmiertechnik II  
Übung 4  
09.06.2016

Maximilian Jenders

## **Hausaufgabe 3 Rückblick**

- Ein Aufzählungstyp (englisch enumerated type) ist ein Datentyp für Variablen mit einer endlichen Wertemenge

```
public static enum InitStrategy {  
    » Zero, Plus, Minus, Random  
}
```

- Enum kann in eigener Klasse oder anderer Klasse sein
- Warum interessant?
- → Beispiel

# Aufgabe 1

- Gefordert: VOLLSTÄNDIGE Zweigüberdeckung
  - Auch Zweige für NAN oder Exceptions
- Aufpassen bei `assertTrue(a==b)` vs `assertEquals(a, b)`
  - Werte/Referenzgleichheit
- `AssertEquals(double d1, double d2, double epsilon)` – warum?
- Test kann `@Test(expected = NullPointerException)` definieren, braucht kein `try/catch`
- `assertEquals(SquareMatrix m1, SquareMatrix m2)` funktioniert nicht – warum?

# Aufgabe 3

- Empirische Messung für Erstellen / Einfügen nicht ausführlich genug (ohne praktisch ermittelte Messwerte) oder gar nicht diskutiert
- Nicht alle 4 Kombinationen zwischen `int/Integer` `int[]/Integer[]`
- Sehr große / kleine empirische Messwerte unlogisch: Kaum bearbeitet
  - Warum?
  - Memory Allocation Pool / Garbage Collection

- Aufgabe 2 gut
- Aufgabe 4 / Zusatz TBD
- Generell: LEST DIE AUFGABENSTELLUNG!
- Sonstige typische Schwierigkeiten?

# Hausaufgabe 4

- Generelles Thema: Algorithmenmuster
  - Greedy, Backpropagation, Divide & Conquer, Dynamische Programmierung
- Nebenbei: Generics (A2)
  - Abstraktion von zu grunde liegenden Typen
  - List<T>
- Beispielsanwendung: Sortierung von Daten, Datentyp an sich ist egal, muss halt immer konsistent sein