

Programmiertechnik II  
Übung 2  
11.05.2016

Maximilian Jenders

- Generell: Gut gelaufen. Durchschnittliche Punkte:
  - Aufgabe 2: 3.56 / 4
  - Aufgabe 3: 3.85 / 4
  - Aufgabe 4: 3.96 / 4
  - Aufgabe 5: 3.85 / 4
  - Aufgabe 6: 3.99 / 4
  - Zusatz: 0.79 / 4
- Gesamtschnitt: 92% der Punkte
- Etwas verzerrt durch „hidden tests“ – Fehler sofort gefunden statt erst in der Korrektur

# Rückblick Übung 1

## Allgemeine Anmerkungen

---

- Tabs/Spaces: Konsistente Einrückungen
- Variablen müssen nicht vorab deklariert werden
  - `String result; (...), result = „test“;` → `String result = „test“;`
- Unnötige temporäre Variablen
  - `str2 = value.toString(); return str2;` → `return value.toString();`
- Kommentare sollten zusammenhängende Blöcke semantisch erklären
  - *// return the accumulated value as string vor return "" + sum*
- Unnötiges else nach throw oder return
  - Benutzt early returns, das erspart Schachtelung!
  - `if (n < 0 || n > 100) { throw new IllegalArgumentException(); }`

# Rückblick Übung 1

## Allgemeine Anmerkungen

---

- Variablennamen sollten mit Kleinbuchstaben anfangen
- `IllegalArgumentException` muss nicht declared werden
- Unnötiges `== true` oder `== false`
- Strings in Länge 1 gesplittet statt chars zu benutzen
- While statt for Schleifen
- Kein Aufbrechen in Funktionen – zu viel Schachtelung

- Aufgabe 2:
  - `Hex(Integer.MAX_VALUE)` führt zu Endlosloop!
- Aufgabe 3:
  - Gute Idee: `sumFiles` ruft nur `sumFiles2` auf
- Aufgabe 4:
  - Häufig wurden unnötig viele Zwischenergebnisse abgespeichert
- Aufgabe 5:
  - `fact(0)` liefert eine `IllegalArgumentException` oder den Wert 0.
  - Anstelle von `while`-Schleifen wurden hin und wieder `for`-Schleifen mit leeren ersten Ausdrücken verwendet (`for(; i < n; i--)`)

- Aufgabe 6:
  - Unzureichende Begründung, weshalb sich die Methode `long iterative(long n)` besonders bei großen Eingaben für `n` deutlich performanter als ihr Gegenstück `long recursive(long n)` verhält -- mitunter wurden lediglich Beobachtungen angestellt.
  - `recursive(0)` liefert keine `IllegalArgumentException`, sondern den Wert 0.
- Zusatz:
  - Falschverstehen von paarweise verschieden (Zählen nur adjazente Paare)

- Genereller Tipp:
  - Schreibt auch eigene Tests. Vor allem bei der Zusatzaufgabe!
- Noch waren wir bei der Bewertung kulant - in Zukunft achtet bitte verstärkt auf die Aufgabenstellung!
  - Beispiele: Falscher Maximalwert benutzt, Fälle  $A=0$  oder  $A > B$  nicht überprüft, durch falschen Logarithmus alle Zehnerpotenzen als Palindrom erkannt...
  - Tests helfen!

- Doppelabgaben:
  - Bitte pro Aufgabe nur von einem Account submitten
  - Bei Doppelabgaben: Abgabe, die den spätesten Submit hatte, wird gewertet



- Generelles Thema: Suchen und Sortieren
- Aufgabe 1: compareTo, Comparator
- Aufgabe 2: Suchen, Aufgabe 3: Sortieren
  - Laufzeitmessungen: openHPI terminiert nach 10s, daher lokal ausführen!
  - Große Datei angeboten, muss nicht benutzt werden
- Zusatzaufgabe: Wortsuche
  - Wichtig: Nicht nur korrekte Lösung wichtig, sondern auch Optimierungen!