

# ExClaim – Digitale Organisation von Programmierübungen

Annette Bieniusa, Sebastian Schweizer, Mathias Weber, Peter Zeller  
AG Software Technology, FB Informatik TU Kaiserslautern

## Was ist ExClaim?

- Automatisierte Gruppeneinteilung nach Präferenz und Freundesgraph
- Bearbeitung von Aufgaben im Team
- Verwaltung von verschiedenen Abgaben-Versionen
- Sichere Testumgebung für eingereichte Lösungen

## Programmieraufgaben – einfach verwalten

Aufgabe	Dateien	Punkte
Aufgabe 1	<div><div>• <b>Minimum.fs</b> ( von Erika Mustermann am Donnerstag, 21.11.2019, 14:44 ) </div><div>Dateien Anzeigen</div><div>Tests starten</div><div>Alle 4 Tests OK!</div><div>Test-Details anzeigen</div><div>Hochladen</div><div>Papierkorb anzeigen/verbergen</div></div> <div><div>✓ Teil a) Minimum ist eine der beiden Zahlen</div><div>✓ Teil b) Minimum ist kleiner gleich der drei Zahlen</div><div>✓ Teil a) Minimum ist kleiner gleich der beiden Zahlen</div><div>✓ Teil b) Minimum ist eine der drei Zahlen</div></div>	6/6
Aufgabe 3	<div><div>• <b>Peano.fs</b> ( von Max Mustermann am Freitag, 15.11.2019, 20:33 ) </div><div>Dateien Anzeigen</div><div>Tests starten</div><div>Es wurden 3 von 5 Tests bestanden. Test-Details anzeigen</div><div>Hochladen</div></div> <div><div>✓ root3: Untere Schranke</div><div>✓ mult3</div><div>✗ search</div><div>✓ exp3</div><div>✗ root3: Obere Schranke</div></div>	6/8

## Für Lernende

- Schnelles Feedback zur funktionalen (teilweise auch nicht-funktionalen) Korrektheit von eigenen Lösungen

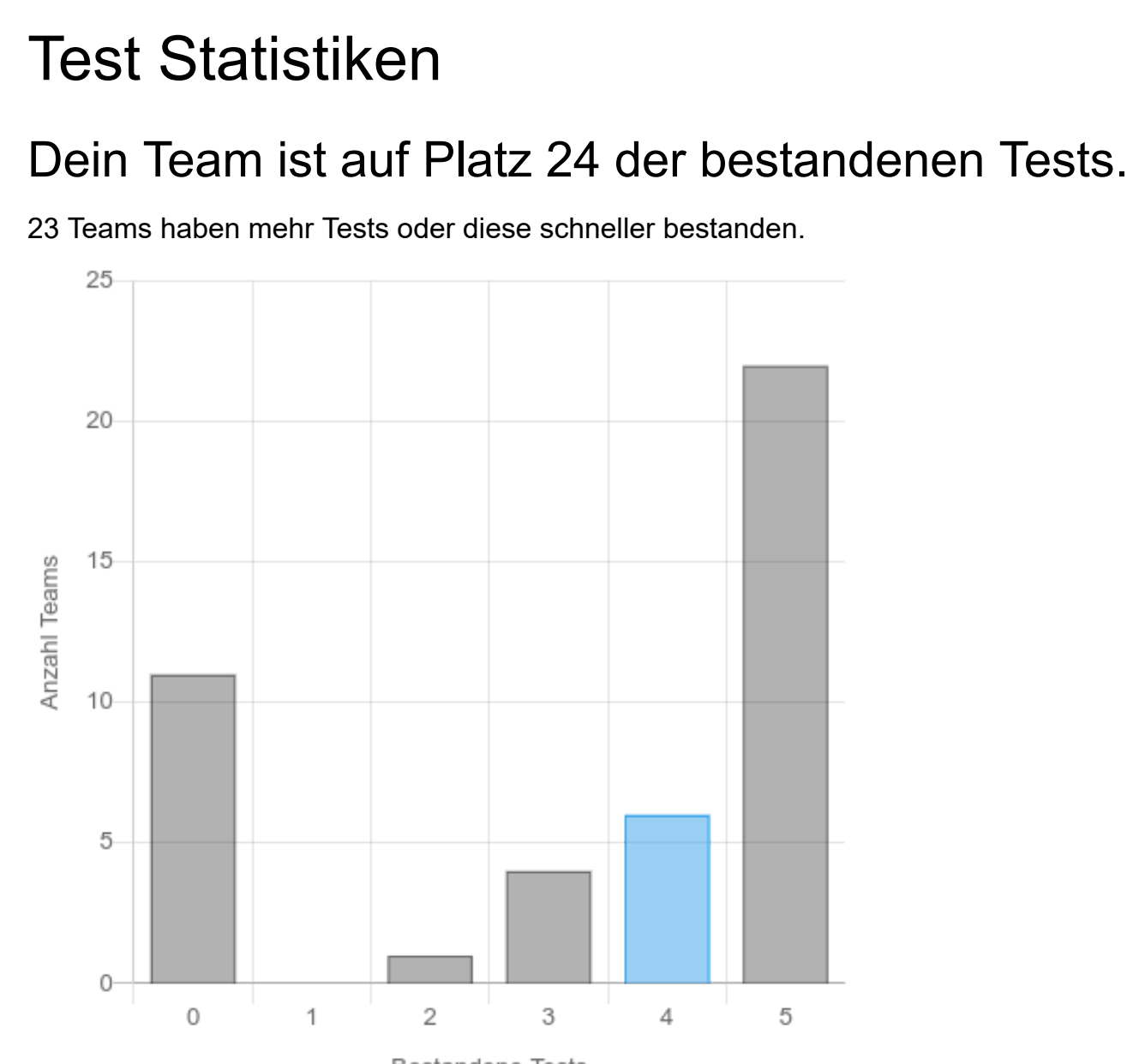
✗ list\_remove1

Error for the following input with parameters 'list\_remove 2':  
4 2 2 15 2 17

Erwartetes Ergebnis:	Tatsächliches Ergebnis:
3 Elemente entfernt: 4, 15, 17	0 Elemente entfernt: 4, 2, 2, 15, 2, 17

- Nicht gebunden an Sprechzeiten, Übungsstunden, etc.
- Hohe Selbstmotivation
- Programmierstil wird durch test-getriebene Herangehensweise positiv geprägt

## Anonymisierter Vergleich mit Peers möglich



## Für Lehrende

- Alles an einem (digitalen) Ort!
- Einsatz von bekannten und unbekannten Tests, die typische Fehler antizipieren
- Einblick in aktuellen Leistungsstand und Verständnis der Studierenden
- Analyse von häufig auftretenden Problemen und Lösungsstrategien
- Referenzsystem erleichtert Korrektur
  - Sicheres Ausführen auf Server
  - Kein Herunterladen, Zwischenspeichern, etc. notwendig
- Verwenden von Templates verringert die Arbeitslast für die Studierenden

## Korrektur durch Annotation von Lösungen

```
16 public int count(boolean[] array) {
17     int counter = 0;
18     for (int i = 0; i < array.length; i++) {
19         if (array[i] == true) {
20             counter++;
21         }
22     }
23     return counter;
24 }
25
```

'if' has incorrect indentation level 16, expected level should be 12.

Avoid unnecessary comparisons in boolean expressions

Weitere Infos

Fall array=null fehlt damit alle Tests funktionieren :)

## Einsatz in Lehrveranstaltungen an der TU Kaiserslautern

- Grundlagen der Programmieren (FB Informatik)
- Software-Engineering 1 (FB Informatik)
- Programmierpraktikum (FB Informatik)
- Programmieren in C (FB ETechnik)
- IT für Maschinenbau (FB Maschinenbau)
- Algorithmen und Datenstrukturen (FB Informatik, Nebenfach)

Aktuell unterstützte Programmiersprachen:

- Java, C, Matlab, F#, Python

## Ausblick

- Zusätzliche Artenopefakte für Testfälle (z.B. Performance-Graph zu Ausführungszeit und Speicherbedarf, Animation der Ausführung)
- Check auf Ähnlichkeit von Lösungen <sup>a</sup>
- Integration mit Authentifizierungsservice wie RHRK Account
- Einbindung in das elektronische Lehrbuch inf-schule, gefördert vom FIT e.V., dem Förderverein Informatik an der TU

<sup>a</sup>Bachelorarbeit "Plagiatsoftware für Programmiercode" von Mario Biedenbach, 2019

## Kontakt

Dr. Annette Bieniusa  
AG Software Technology  
FB Informatik, TU Kaiserslautern  
bieniusa@cs.uni-kl.de



Danke an alle Beteiligten, insbesondere an unsere Hiwis Daniel Jenet und Talant Asankozhoev!

Gefördert als TUK LehrePlus Projekt 2018