Fragen, Antworten, Kommentare zur aktuellen Vorlesung

Die Online-Befragung zur genutzten alternativen Veranstaltungsform und zur Lehrevaluation ist online. Bitte ausfüllen: https://forms.gle/fZTBWPmUhK4oqLHw5. Sie werden eventuell aufgefordert sich bei Google anzumelden, das ist nur notwendig, wenn Sie in der Bearbeitung eine Pause machen und das Teilergebnis zwischenspeichern wollen. Die Befragung endet am 19.12., die Ergebnisse stehen in einem nachfolgenden Fragen&Antworten-Dokument auf der Webseite der Veranstaltung.

Das letzte Aufgabenblatt 11, das abgenommen wird, ist online. Wenn das Blatt abgenommen wurde, besteht keine Verpflichtung mehr zur Praktikumsteilnahme. Die Praktikumsleiter sind natürlich trotzdem erreichbar.

Frage: Können oder sollen wir nicht Eclipse, Git und CodeTogether nutzen?

Dies sind sehr wichtige Werkzeuge im Laufe des Studiums, aber aus meiner Sicht frühestens ab dem zweiten Semester. Generell haben meine Erfahrungen in ersten Semestern gezeigt, dass jedwedes nützliche, aber kompliziertere, Werkzeug dazu führt, dass unsichere Studierende noch unsicherer werden, da sie zusätzlich das neue Tool verstehen müssen und so oft nicht zum Mehrwert kommen. Jede kleine zusätzliche Stufe führt zum Verlust von Personen am Anfang des Studiums.

Eclipse schauen wir kurz am Ende der Vorlesung an. BlueJ ist das einzige Werkzeug mit dem ein einfacher sauberer Einstieg in die Objektorientierung funktioniert, was bei Studierenden zu besseren Lernerfolgen führt (Erfahrungen der Entwickelnden von BlueJ und auch meine Erfahrung). IntelliJ sieht besser aus, kann im Wesentlichen nicht mehr als Eclipse und erfüllt die Installationsanforderungen für die identische Form für die Hochschule und KleukersSEU nicht.

Git ermöglicht es systematisch verschiedene Versionen der gleichen Software zu verwalten und zugänglich zu machen. Das unterstützt einzelne Personen und Gruppenarbeiten mit gemeinsamem Git-Zugang. Git steht auf meiner Empfehlungsliste zum Selbststudium zwischen erstem und zweitem Semester.

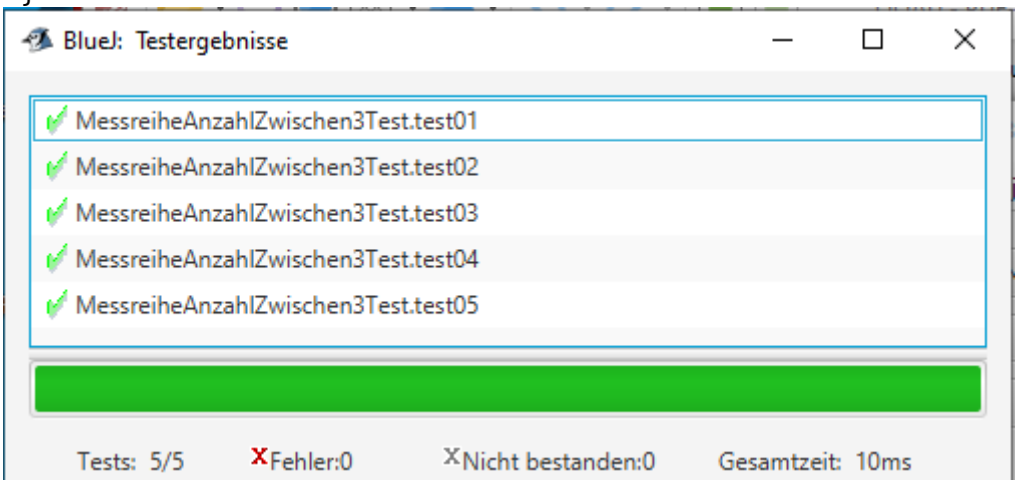
CodeTogether ermöglicht, dass mehrere Personen gemeinsam in einem Editor Programmcode lesen und parallel bearbeiten können (live sharing). Das ist sehr cool und kann bei einer organisierten Zusammenarbeit sehr interessant sein. CodeTogether arbeitet dabei z. B. über Werkzeuggrenzen hinaus und eine Teilnahme im Browser kann sogar möglich sein. Ein Einführungstext steht in https://dzone.com/articles/remote-pair-programming-with-intellij-eclipse-and. Wieder ein tolles Programm, wenn alle Personen ein gewisses Programmier-Niveau erreicht haben. In den ersten beiden Semestern steht bewusst aber der Kampf „ein Mensch und eine Programmieraufgabe“ im Mittelpunkt und soll durch Gruppenarbeiten nur sinnvoll flankiert werden. Es ist auch möglich das Eclipse in der KleukerSEU um weitere PlugIns zu erweitern.

Erinnerung: Tests helfen bei Entwicklung und ihre Nutzung ist in größeren Projekten ein Standard. Ob Tests vor oder nach der Entwicklung geschrieben werden, hängt wieder von Rahmenbedingungen ab. Generell muss man sich dazu merken, dass Tests notwendig (man muss sie machen) aber nicht hinreichend für eine gute Software-Qualität sind. Dies bedeutet, dass es trotz vollständig erfüllter Tests immer noch massive Fehler in den Programmen geben kann, siehe auch [Kle19], über die

Bibliothek kostenlos herunterladbar. Dies bedeutet für Ihre Praktika, dass Ihre Programme trotz einer recht hohen Testanzahl noch falsch sein können und Sie Ihre Algorithmen weiterhin im Kopf überprüfen müssen. Wenn sowas nebenbei passiert, also falsche Programme mit laufenden Tests, schicken Sie mir solche Programme gerne zu, da ich dann zumindest die Testanzahl erhöhen kann (was nie hinreichend sein wird).

Ein sehr interessantes Beispiel ist diese vermeintliche Lösung zur Aufgabe 28 g)

```
public int anzahlZwischen3(int min, int max) {
  if (this.messwerte == null || this.messwerte.isEmpty()) {
    return 0;
  }
  int ergebnis = 0;
  for (int i : this.messwerte) {
    if (this.messwerte.get(i-1) <= max && this.messwerte.get(i-1) >= min){
      ergebnis = ergebnis + 1;
    }
  }
  return ergebnis;
}
```



Es sollte klar sein, dass hier die ForEach-Schleife nicht richtig verstanden und i nicht als Element, sondern als Index angesehen wurde. Da es mit get(i) eine Exception gibt, wurde mit i-1 experimentiert und siehe da, es klappt auch bei den korrigierten (siehe oben) Tests.

Den Fehler erkennt man mit Programmiererfahrung sehr schnell, aber Sie kommen immer wieder in Situationen, in denen Sie Anfänger auf einem Gebiet sind, so dass ein vergleichbarer Fehler dann auch möglich ist. Deshalb benötigt man z. B. beim Selbststudium immer einen erfahreneren Coach, der zumindest am Ende sich Ergebnisse anschaut.

[Kle19] S. Kleuker, Qualitätssicherung durch Softwaretests, 2. aktualisierte und erweiterte Auflage, Springer Vieweg, Wiesbaden, 2019