

Übungsblatt 8 – Ausgabe am 15.06.2020

Abgabe bis 22.06.2020, 12 Uhr, per Mail

Aufgabe 8.1: Grundlagen Statische Analyse

Beantworten Sie die folgenden Fragen so kurz wie möglich und so ausführlich wie nötig.

- Nennen Sie die sieben Arten der statischen Programmanalyse, die in der Vorlesung erwähnt werden und beschreiben Sie die Verfahren jeweils in einem Satz.
- Welche „psychologischen“ Probleme können bei der Qualitätssicherung von Software auftreten und was sind mögliche Gegenmaßnahmen?
- Ist die folgende Schlussfolgerung aus dem Satz von Rice korrekt? Begründen Sie!
Es ist unmöglich, zu entscheiden, ob eine nichttriviale Eigenschaft für ein Programm gilt, das in einer Turing-mächtigen Programmiersprache geschrieben ist.
- Warum spielen Kontrollflussanomalien in modernen Programmiersprachen wie Java de facto keine Rolle mehr?
- Was bezeichnet den Grad der Kohäsion und Kopplung in einem objektorientierten Programm?

Aufgabe 8.2: Metriken

Berechnen Sie die untenstehenden Metriken für die nachfolgend abgebildete Methode. Dokumentieren Sie den jeweiligen Rechenweg und geben Sie auch an, mit welcher Skala (Nominal-, Ordinal- oder Rationalskala) Sie die jeweilige Metrik messen.

```
01  public static double[] getMinMax(float[] a) {
02      double min = Double.MAX_VALUE;
03      double max = -Double.MAX_VALUE;
04      double value;
05      for (int i=0; i<a.length; i++) {
06          value = a[i];
07          if (value<min)
08              min = value;
09          if (value>max)
10              max = value;
11      }
12      double[] minAndMax = new double[2];
13      minAndMax[0] = min;
14      minAndMax[1] = max;
15      return minAndMax;
16  }
```

- Lines of Code
- Zyklomatische Zahl (nach McCabe)
- Programmlänge (L) nach Halstead
- Programmgröße (V) nach Halstead
- Live Variables

Anmerkung:

1. Stellen Sie Ihre Berechnungsschritte nachvollziehbar dar. Geben Sie hierfür gegebenenfalls zusätzlich Tabellen, attribuierte Kontrollflussgraphen etc. mit an.
2. Geben Sie für die Teilaufgaben c) und d) zusätzlich die Liste betrachteter Operatoren und Operanden mit an. Für Operanden gilt wie üblich, dass **jede Verwendung** pro Anweisung **einzeln** gezählt wird.