MASTER OF SCIENCE IN ENGINEERING

Forschungsprojektarbeit in der MRU Computer Science / Fachgebiet ICT

Automatische Feedbackgenerierung für Java-Programme

Ausgangslage

Mit der Einführung von weltweit offenen Online-Lehrangeboten (MOOCs) sind bei deren Anbieter unterschiedliche Herausforderungen bezüglich der Skalierung entstanden. In Einführungskursen ins Programmieren mussten auf einmal tausende Computerprogramme anstatt hunderte wie bisher bei einer einzigen Übung begutachtet, kommentiert und allenfalls bewertet werden. Daraus entstand der Bedarf einer automatischen Feedbackgenerierung, um die limitierten Personalressourcen auf Fälle anzusetzen, welche nicht automatisch abgehandelt werden können.

Eine solche automatische Feedbackgenerierung bietet sich auch an der Hochschule für Technik in den Einführungskursen ins Programmieren an, um die Übungsbetreuung des begleiteten Selbststudiums bei wachsenden Studierendenzahlen auf einem hohen Niveau halten zu können. Die studentische Bearbeitung von Programmierübungen findet oft zu Randzeiten oder an Wochenenden statt. Damit die Studierenden auch dann in Sekundenschnelle ein qualitativ hochstehendes Feedback erhalten, braucht es ein ausgeklügeltes automatisches Feedbacksystem, welches mehr zu leisten vermag, als nur die Korrektheit bzw. Fehler bei verschiedenen Testbeispielen zurückzumelden.

Zielsetzung / Methodik / Vorgehen

Verschiedene Ansätze für die Entwicklung solcher Feedbacksysteme sind bekannt und sollen evaluiert und den Bedürfnissen der Hochschule für Technik entsprechend umgesetzt werden. In den uns bekannten Feedbacksystemen wird jeweils von einer korrekten Musterlösung ausgegangen, die zu jeder Programmieraufgabe vorliegen muss.

Das zu entwickelnde Feedbacksystem soll primär auf Java ausgerichtet sein und kleinere Programme, wie sie im einführenden Programmierunterricht vorkommen, und in vier Stufen analysieren:

- syntaktische Korrektheit,
- Korrektheit,
- Effizienz,
- Einfachheit.

Auf allen Stufen sollen ausführliche Analysen und Ist-Soll-Vergleiche mit der hinterlegten Musterlösung durchgeführt werden. So soll beispielsweise auf der ersten Stufe die syntaktische Korrektheit des studentischen Programms mithilfe des Java-Compilers und ausführlicheren Fehlerbeschreibungen, wie sie in Web-Ressourcen auffindbar sind, erreicht werden. Auf Stufe zwei und drei werden anhand der Musterlösung automatisch Testbeispiele generiert, welche zur Überprüfung der Korrektheit und der Effizienz der studentischen Programme eingesetzt werden. Zudem können auch Ansätze der Programmverifikation zum Einsatz kommen. Auf Stufe vier schliesslich dienen Programmstrukturanalysen wie Iterationsarten, Verschachtelungstiefen oder Methodenarten (Instanz- vs. Klassenmethoden) zur Untersuchung der Einfachheit der eingereichten Lösungen.

Verlangte Skills: Sehr gute algorithmische Programmierkenntnisse

Teilaufgabe für den Masterstudierenden

Das Projekt kann in 3 Stufen durchgeführt werden: Fortlaufende Projekte 7 (syntaktische Korrektheit), 8 (Korrektheit) und 9 (Effizienz und Einfachheit)

Studienart: [x] Vollzeitstudium
[x] Teilzeitstudium 50% mit Assistenzanstellung

Projektorganisation: Einzelarbeit

Projektfinanzierung: (Lehrfonds FHNW)

Arbeitsort: Windisch

Advisor: Prof. Dr. Christoph Stamm