

Portfolio zur Vorlesung Programmierung II

Sven Fiergolla, 1252732

22. August 2017

Einführung

Für das Portfolio zur Vorlesung Programmierung II war es uns freigestellt, die Aufgaben, welche in den Onlineübungen bearbeitet wurden, zusammenzustellen. Im folgenden erläutere ich meine Aufgabenauswahl.

Onlineübung 1 und 2 als Wiederholung beziehe ich komplett mit ein, diese Themen habe ich gut verstanden und konnte das Gelernte umsetzen. Von Onlineübung 3 beziehe ich nur die 1. Aufgabe (Netzwerkprogrammierung) da ich mehrere Probleme mit Aufgabe 2 (Nebenläufigkeit) hatte und die Aufgabe dementsprechend ausgefallen ist. Onlineübung 4 beziehe ich ebenfalls ganz mit ein, die Einführung in funktionales Programmieren war recht verständlich. Von der großen Onlineübung beziehe ich nur den Java-Teil mit ein (Aufgabe 1-3) da ich mich darauf am konsequentesten vorbereitet habe und mein Verständnis in diesem Aufgabenbereich am solidesten ist.

Mein Hauptbeweggrund für die Auswahl dieser Aufgaben war logischerweise das Erreichen des Maximums der aufsummierten Punkte. Gleichzeitig habe ich in diesen Aufgaben das solideste Wissen, welches sich durch die erreichten Punkte widerspiegelt. Die Aufgaben die ich nicht mit einbezogen habe (die Aufgabe zur Nebenläufigkeit und die Skala-Aufgaben der großen Onlineübung) waren auch die Aufgabenbereiche in welchen ich mich am unsichersten fühlte.

Aufgabenauswahl Portfolio Programmierung II Sommersemester 2017

Der von Ihnen auszufüllende Bereich ist rot markiert: Bitte tragen Sie in der Spalte "Erreichte Punkte" alle erreichten Ergebnisse aus den jeweiligen Onlineübungen ein und markieren Sie anschließend mit "x" (kleiner Buchstabe x) die Aufgaben, die Sie einbringen möchten. Sie können außerdem eine Aufgabe auf 5 Punkte abstufen, um die Zielpunktzahl zu erreichen. Die entsprechende Aufgabe können Sie mit einem "x" in der Spalte "Abgestuft" markieren. Falls Sie an einer der Onlineübungen entschuldigt gefehlt haben, können Sie die entsprechenden Punkte aus der Spalte "Erreichbare Punkte" löschen, dann aktualisiert sich die Zielpunktzahl entsprechend. Bitte denken Sie auch daran in der ausgedruckten Fassung die beigefügte Erklärung zu unterschreiben.

	Erreichbare Punkte	Erreichte Punkte	Ausgewählt	Abgestuft	Auswahl	
					Eingebrachte Punkte	Erreichte Punkte
Onlineübung 1						
Aufgabe 1	10	7	x		10	7
Aufgabe 2	10	7	x		10	7
Onlineübung 2						
Aufgabe 1	5	3	x		5	3
Aufgabe 2	10	9	x		10	9
Aufgabe 3	5	2	x		5	2
Onlineübung 3						
Aufgabe 1	10	10	x		10	10
Aufgabe 2	10	2			0	0
Onlineübung 4						
Aufgabe 1	15	12	x		15	12
Aufgabe 2	5	5	x		5	5
Große Onlineübung						
Aufgabe 1	5	2.5	x		5	2.5
Aufgabe 2	10	8	x		10	8
Aufgabe 3	10	9	x		10	9
Aufgabe 4	5	0			0	0
Aufgabe 5	10	4			0	0
Maximum:	120					
Ziel:	95				95	74.5 78.42%

Hiermit erkläre ich, dass ich das Portfolio selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe (siehe §19 (7) APO Bachelor Universität Trier).

Datum

Unterschrift

Reflexion

Zu Beginn der Vorlesungsreihe beschäftigten wir uns mit einfachen Datenstrukturen und deren Aufbau, beispielsweise verkettete Listen, binäre Bäume und Schlangen als Wiederholung. Da diese Strukturen als Grundlage für viele der weiterführenden Themen sind, ist die Behandlung dieser notwendig und hilfreich.

Ebenso wurde die Funktionsweise von *Interfaces*, *Streams* und *Exceptionhandling* rekapituliert und den Studenten wurde das *log4j*-Framework zum standardisierten Logging der eigenen Anwendungen erläutert und dessen Verwendung nahegelegt. Mit *Streams* wurde uns die Möglichkeit zur funktionalen Programmierung in Java gegeben, ein Konzept, was im weiteren Verlauf der Vorlesung, durch die Verwendung von Scala, als funktionale Sprache, gefestigt wurde. Anständiges *Exceptionhandling* und *Logging* wurde zwar gelehrt, die wirkliche Bedeutung dieser Konzepte versteht man jedoch erst, wenn man eigene, größere Projekte verfasst, welche ohne jene Konzepte praktisch nicht umzusetzen sind.

Anschließend beschäftigten wir uns mit graphischen Benutzeroberflächen unter Gebrauch des GUI-Toolkits *SWING* und erstellten eigene Benutzeroberflächen oder Spiele (Mienenfeger aus der Vorlesung). Auch dies Thema ist von großer Bedeutung, da die meisten Endbenutzeranwendungen nicht auf eine graphische Oberfläche verzichten können, um auch von unerfahrenen Anwendern genutzt werden zu können.

In den folgenden Wochen beschäftigten wir uns mit *Nebenläufigkeit* in Form von mehreren Threads in einer Anwendung. Dazu animierten und simulierten wir mehrere Kugeln auf einem Feld, welche miteinander kollidierten. Das Debuggen parallelisierter Anwendungen stellte sich häufiger als Herausforderung heraus, da *Deadlocks* und *race conditions* schwierig zu finden sind. Dennoch ist die nebenläufige Programmierung vorallem bei rechenintensiven und zeitkritischen Problemen vonnöten.

Unter Verwendung von *UDP/TCP* als Netzwerkprotokolle, lernten wir das Programmieren verteilter Anwendungen bzw. Netzwerkprogrammierung. Im Zuge dessen benutzen wir ebenfalls die Konzepte *Serialisierung* und *Reflection* um Java-Objekte übertragen zu können und mit diesen zu arbeiten. Als Anwendung in diesem Bereich programmierten wir das Spiel „Pong“ als Client-Server-Anwendung.

Abschließend machten wir uns mit dem Konzept der funktionalen Programmierung vertraut und lernten dazu die Sprache Scala. Dazu arbeiteten wir mit einfachen Datenstrukturen wie Listen und Bäume und nutzen *Pattern Matching* oder *Currying*. Funktionale Programmierung als neues Programmierparadigma wirft eine neue Blickrichtung auf das Lösen von Problemen, da die Programme eine reine Folge von Funktionen sind. Vorallem für aktuelle Bereiche der Programmierung wie zb. neuronale Netze sind häufig funktional gelöst, was dieses Thema sehr interessant macht.

Zusammenfassend lässt sich sagen, dass ich in der Vorlesungsreihe Programmierung II die wesentlichen Konzepte und Werkzeuge zum Schreiben komplexer, objektorientierter Software, sowie die Grundlagen funktionaler Programmierung gelernt habe.