

Softwareentwicklungspraktikum Android-Programmierung

Organisation & Einführung

Lorenz Schauer
Lehrstuhl für Mobile und Verteilte Systeme
Sommersemester 2017



■ Voraussetzungen:

- Erfolgreiche Teilnahme an der Veranstaltung „Einführung in die Programmierung“
- Grundverständnis von objektorientierten Programmiersprachen (bspw. Java)
- Erste praktische Programmiererfahrung wünschenswert

■ Ziele:

- Selbstständig Programmieren lernen in der Praxis und im Team
 - Mobile Plattform Android
 - Linux-Server
 - Verwaltung und Versionskontrolle mittels Git
- **Eigenverantwortliches Arbeiten**
- Konzeption eines größeren Softwareprodukts (Spiel)
- Projektmanagement
- Präsentation der Ergebnisse (am Ende des Kurses)

■ Zielgruppe:

- Bachelorstudenten mit Hauptfach Informatik/Medieninformatik

▪ Grundsätzlicher Ablauf:

- Wöchentliche Theorieveranstaltung:
 - Dienstags von 10.00 – 12.00 Uhr
 - Hauptgebäude, Geschwister-Scholl-Pl. 1, Raum A 014
- Wöchentliches Tutortreffen:
 - Mit der kompletten Gruppe und dem jeweiligen Tutor (ca. 1 Stunde)
 - Ort und Zeit ist in gemeinsamer Absprache in der 1. Vorlesungswoche zu bestimmen!
- Eigenständiges Arbeiten!!!

▪ Kurswebseite (wichtig):

- <http://www.mobile.ifi.lmu.de/lehveranstaltungen/sep-sose17/>
- Aufgaben, Meilensteine und alle wichtigen Informationen

▪ Gruppeneinteilung:

- Die Gruppeneinteilung ist abgeschlossen.
- Jeder Teilnehmer sollte bereits eine Email (uniworx hinterlegte Adresse) mit den entsprechenden Informationen bekommen haben?!?
- Bitte findet euch selbstständig und möglichst bald (in dieser Woche) für ein erstes Treffen mit eurem Tutor!

Vergütung/Erwartung

- Modul WP 1*

- **12 ECTS**

- 1 ECTS = 30 Arbeitsstunden pro Semester

⇒ $12 * 30 = 360$ Stunden im Semester pro Person

⇒ Bzw. $360 \text{ Std} / 14 \text{ Wochen} = 25,71$ Stunden p.P. pro Woche

⇒ $360 * 5 = \mathbf{1.800 \text{ Stunden}}$ für das Projekt insgesamt!

Mögliche Zusammensetzung:

Veranstaltung	Zeitaufwand (ca.)
Plenum	1,5 Stunden
Tutortreffen	Ca. 1 Stunde
Gruppentreffen	Ca. 8 Stunden
Eigenständiges Arbeiten	Ca. 15 Stunden

*Studienordnung für den Bachelorstudiengang Informatik bzw. Medieninformatik

Was wird bewertet?

- Mitarbeit im Team
 - Bei Tutor- und Gruppentreffen
- Mitarbeit und Anteil am Projekt
 - Ideen, Code, Hilfestellungen...
 - Überprüfung durch Git-Repositories und Code-Review am Ende des Kurses
- Vorprojekt
 - Jeder einzeln!
- Hauptprojekt
 - Präsentation
 - Codereview
- Generell wird eine **individuelle Leistungsbewertung** durchgeführt!

Alle Abgaben müssen **innerhalb der gesetzten Fristen** erfolgen!

- Sollte ein Teilnehmer sich „durchmogeln“ wollen, gibt es Einzelgespräche und Sanktionen bis hin zum Ausschluss aus dem Praktikum

Datum	Thema	Aufgabe/Meilenstein
25.04.2017	Einführungsveranstaltung	Gruppenfindung, Kennenlernen, Termine für Treffen ausmachen und 1. Treffen. Infrastruktur aufsetzen und Tools installieren (Git, Android Studios etc.)
02.05.2017	Android Basics	Blatt 1: Activities, Fragments, Services, Content Providers und Receivers Deadline: Sonntag, 07.05.2017 23:59:00
09.05.2017	RESTful Webservices	Blatt 2: Umgang mit REST Deadline: Sonntag, 14.05.2017 23:59:00

Insgesamt 3 Wochen

- Einzelabgaben über Uniworx
- Aber: Hilfestellung durch Gruppe und Tutor

Datum	Thema	Aufgabe/Meilenstein
16.05.2017	Beginn des Hauptprojekts (Kick-Off) Spielregeln werden bekannt gegeben	Management des Hauptprojekts, Aufgaben verteilen, Spielkonzept mit Tutor besprechen!
23.05.2017	Konzeption, Architektur & Datenmodell	Konzept mit Tutor besprechen
30.05.2017	Client/Server, Schnittstellen	Features V0.1 umsetzen
06.06.2017	Implementieren u. Testen	Features V0.2 umsetzen
13.06.2017	Implementieren u. Testen	Features V0.3 umsetzen
27.06.2017	Implementieren u. Testen	Features V1 umsetzen
11.07.2017	Test-Phase	Beta-Tests Abgabe des Test-Protokolls bis: 16.07.2017
18.07.2017	Debugging-Phase	Alle Fehler beheben
30.07.2017	Abschluss	Projektabgabe! Deadline: Sonntag, 30.07.2017 23:59:00
Ab 31.07.2017	Präsentation	Projekt-Präsentationen + Code-Review

Insgesamt 11 Wochen

Zusammenfassung der wichtigsten Termine (Abgaben)

- Die genannten Deadlines sind unbedingt einzuhalten!
- Es werden **keine Ausnahmen** gemacht!

Aufgabe	Ausgabe	Abgabe	Bemerkung
Übungsblatt 1 (Basics)	Mo, 01.05.2017, 8:00 Uhr	So, 07.05.2017, 23:59 Uhr	Ausgabe und Abgabe erfolgt über Uniworx.
Übungsblatt 2 (REST)	Mo, 08.05.2017, 8:00 Uhr	So, 14.05.2017, 23:59 Uhr	Ausgabe und Abgabe erfolgt über Uniworx.
Hauptprojekt	Di, 16.05.2017	So, 30.07.2017, 23:59 Uhr	Hauptprojekt wird in der Theorieveranstaltung am 16.05.2017 vorgestellt. Abgabe erfolgt über Git (Master-Branch, letzter Push vor der Deadline)

Die nächsten Schritte (am besten diese Woche):

- **Gruppe finden** und kennenlernen mit Tutor
- **Tools installieren** und ausprobieren:
 - Android Studios: <https://developer.android.com/studio/index.html>
 - WarmUp: Bspw. HelloWorld-App zum Laufen bringen.
- **Git einrichten:**
 - Gitlab Kennung bei der RGB beantragen:
 - Infos unter: <http://www.rz.ifi.lmu.de/Dienste/Gitlab.html>
 - Einen „Teamleiter“ pro Gruppe bestimmen der die *Master*-Rolle übernimmt
 - Nur der **Teamleiter** darf auf den Master-Branch pushen und diesen mergen!
 - Wenn alle Teilnehmer der Gruppe einen Gitlab-Zugang haben, dann Mail an lorenz.schauer@ifi.lmu.de mit Gruppennummer, vollständigen Namen der Mitglieder und Kennzeichnung des Teamleiters.
 - Ihr bekommt daraufhin euer Git-Repository zur Entwicklung des Hauptprojekts über Mail mitgeteilt. Bitte prüfen, ob alles geht!
- **Auf Linux-Server anmelden** und nachschauen, ob alles geht
 - Nutzernamen und Passwort bekommt ihr in der Mail bzgl. Git mitgeteilt.

Euer zentrales Git-Repository liegt dann auf:

https://gitlab.cip.ifi.lmu.de/sep_android_sose17/<name>

- Auf lokales Gerät Kopieren (bspw.: `git clone <remote-url>`)
- Auf dem lokalen Repository arbeiten:
 - `git add <file>`
 - Merkt Änderungen an der Datei für den nächsten Commit vor (Staging Area)
 - Mit `git status` lässt sich der Zustand von Arbeitsverzeichnis und Staging Area überprüfen
 - `git commit -m „text für History“`
 - Übernimmt die vorgemerkten Änderungen in das lokale Repository
- Änderungen bekannt machen
 - `git push`
 - Übermittelt die Commits des lokalen Repos in das zentrale Repo
- Änderungen ins lokale Repository übernehmen:
 - Entweder über `git fetch <remote>` und dann `git merge origin/<current-branch>`
 - Oder in einem Befehl über `git pull <remote>`

Möglichkeit zum Arbeiten mit Branches:

- 2 Haupt-Banches existieren im Repository:
 - origin/master
 - Nur vom Teamleiter veränderbar
 - Soll immer das Lauffähige Produkt enthalten!
 - origin/develop
 - Hier wird entwickelt
 - Soll damit den letzten Entwicklungsstand enthalten
 - Wird bei einem stabilen Stand *released*
 - Push in den Master vom Teamleiter

Beispiel: Ein möglicher Ablauf:

- 2 Team-Mitglieder Alice und Bob (Teamleiter)
- Alice entwickelt lokal im develop-Branch
 - Commit der Änderungen und push des develop-Branch
 - Alice gibt Bob Bescheid, dass Feature fertig ist.
- Bob befindet sich lokal im master-branch
 - Bob holt sich den aktuellen develop-Stand
 - `git checkout develop` und dann `git pull`
 - Bob merged den develop-Branch in seinen lokalen master-Branch
 - Bsp.: `git checkout master` und dann `git merge develop`
 - Bob test, ob alles funktioniert
 - Wenn ja, dann Release
 - `git push`
- Ggf. sinnvoll, wenn jedes Team-Mitglied auf seinem eigenen Branch entwickelt und fertige Features in den develop-Branch überführt.

Zum Thema eigenständiges Arbeiten:

- Zu jedem Problem gibt es tonnenweise Information (Google ist dein Freund)
- Diese nützlichen Links sind auch auf der Kurswebseite angegeben:
 - Zur Android-Entwicklung:
 - <https://developer.android.com/index.html>
 - <http://www.programmierenlernenhq.de/android-tutorials/>
 - <http://docs.oracle.com/javase/8/docs/api/>
 - Zum Arbeiten mit Git:
 - <https://de.atlassian.com/git/tutorials>
 - <http://nvie.com/posts/a-successful-git-branching-model/>
 - <https://www.kernel.org/pub/software/scm/git/docs/>
 - <https://www.kernel.org/pub/software/scm/git/docs/user-manual.html>

Zu allererst an den jeweiligen Tutor! Kontaktadresse wurde per Mail verschickt.

- Arian Weber
- Robert Müller

Sonstige Unklarheiten (Inhalt, Bewertung):

- Dr. Michael Beck (michael.beck@ifi.lmu.de)
- André Ebert (andre.ebert@ifi.lmu.de)

Bei organisatorischen bzw. technischen Problemen (Gitlab, Server, usw.):

- Lorenz Schauer (lorenz.schauer@ifi.lmu.de)