

Einzelbeispiel

UE Software Engineering II - SS2023

1. Vorbereitungen

Stellen Sie sicher, dass Sie die folgenden Vorbedingungen erfüllen:

- Git installiert
- GitHub Account erstellt
- Öffentlich zugängliches Git Repository für das Einzelbeispiel angelegt
- Android Studio installiert
- Android SDK installiert (+evtl. Emulatoren)

Sollten Sie noch keine Programmiererfahrungen mit Android gemacht haben, lesen Sie sich in die grundlegenden Konzepte von Android ein (Siehe auch die untenstehenden Links). Da auch in der Gruppenphase für Android Umgebungen entwickelt wird, ist es sehr wichtig, dass Sie die grundlegenden Konzepte von Android beherrschen.

2. Projektbeschreibung

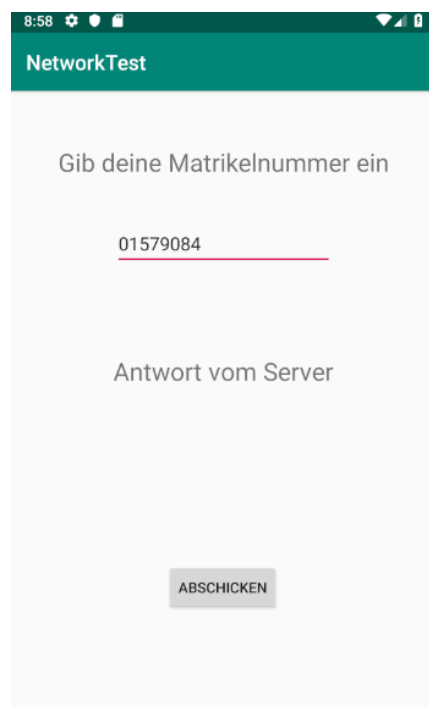
2.1

Erstellen Sie eine einfache Android-Anwendung, die ein Eingabefeld, ein Textfeld und einen Button enthält. Das Eingabefeld soll nur numerische Eingaben erlauben. Durch Tap auf den Button soll die Eingabe via TCP an einen Server geschickt werden. Sobald eine Antwort vom Server eintrifft, soll diese am Bildschirm erscheinen.

Der Server nimmt eine Matrikelnummer als Bytestream über die TCP Verbindung entgegen, führt eine Berechnung aus und sendet das Ergebnis zurück.

Die Server-Domain lautet: **se2-isys.aau.at**, Port: **53212**.

Das User Interface könnte beispielsweise wie folgt aussehen:



2.2

Erweitern Sie die Applikation aus 2.1 entsprechend Ihrer Matrikelnummer. Berechnen Sie dazu den Wert Matrikelnummer Modulo 7 und lösen sie die dazugehörige Aufgabe aus der Tabelle. Das Ergebnis soll wieder am Bildschirm ausgegeben werden.

Nutzen Sie dafür das existierende Nummernfeld und Textfeld für Ein- und Ausgabe. Fügen Sie außerdem noch einen weiteren Button ein, der die Berechnung auslöst.

Matrikelnummer mod 7 = __	Aufgabe (Input jeweils MatrikelNr)
0	Nur jene Ziffern ausgeben, die Primzahlen sind
1	Alternierende Quersumme bilden und ausgeben, ob diese gerade oder ungerade ist
2	Matrikelnummer sortieren, wobei zuerst alle geraden, dann alle ungeraden Ziffern gereiht sind (erst die geraden, dann alle ungeraden Ziffern aufsteigend sortiert)
3	Prüfen, ob zwei beliebige Ziffern existieren, die einen gemeinsamen Teiler > 1 haben. Werden zwei Ziffern mit gemeinsamem Teiler gefunden, soll deren Index ausgegeben werden
4	Jede zweite Ziffer der Matrikelnummer durch ASCII Characters ersetzen, wobei 1 = a, 2 = b, ...
5	Quersumme der Matrikelnummer bilden und anschließend als Binärzahl darstellen
6	Ziffern der Größe nach sortieren, Primzahlen werden gestrichen

Beispiel: Ihre Matrikelnummer lautet 1160876. Modulo 7 ergibt das 3, somit muss die Aufgabe 3 implementiert werden.

3. Abgabe

Das Abgabeende ist am 12.03.2023 um 23:55. Verwenden Sie zur Implementierung unbedingt **Git** als Source Code Management System. Committen Sie alle Einzelschritte und achten Sie darauf, dass Sie Ihren Code auch auf das **GitHub Repository** pushen. Commits, die nach der Deadline eingebracht werden, können nicht mehr berücksichtigt werden.

Abgabe Einzelbeispiel: Um Ihr Projekt abzugeben, füllen Sie das `abgabe.xml` File mit Ihren Daten und geben Sie es **syntaktisch korrekt** vor Ablauf der Deadline auf Moodle ab. Zum Beispiel:

```
<person>
  <name>Max Mustermann</name>
  <matrikelnummer>9914625</matrikelnummer>
  <lastcommithash>49ea5d61c3fe2edb01dce289cdb703850d359ba8</lastcommithash>
  <githubusername>MaxMusterGitHub</githubusername>
  <repositoryname>SE2Einzelphase</repositoryname>
  <repositoryurl>
    https://github.com/MaxMusterGitHub/SE2Einzelphase
  </repositoryurl>
</person>
```

4. Nützliche Links

Bei Fragen nutzen Sie das Moodle Forum der Einzelphase. Es wird sowohl von unseren Tutoren als auch von der LV-Leitung betreut.

Git

<https://rogerdudler.github.io/git-guide/index.de.html>

<https://cbea.ms/git-commit/>

<https://git-scm.com/docs/gittutorial>

<http://www.vogella.com/tutorials/Git/article.html>

<https://education.github.com/git-cheat-sheet-education.pdf>

Android

<https://developer.android.com/training/index.html>

<https://developer.android.com/guide/components/activities/activity-lifecycle.html>

Netzwerk

<https://medium.com/hootsuite-engineering/asynchronous-android-programming-the-good-the-bad-and-the-ugly-c48a0b73665f>