

Lern- und Arbeitsauftrag 1305 Codes, NoSQL und Backend

Titel:	Modul 165, 306 - Gruppenarbeit
Modul:	IMS-Lernatelier 2 Informatiker/in EFZ
Autor / Version:	Michael Schneider
Sozialform	Gruppenarbeit mit 1 bis 4 Personen
Hilfsmittel:	<ul style="list-style-type: none"> • Alle

Ausgangslage

In diesem Auftrag sollen folgende Module vertieft werden:

165 NoSQL-Datenbanken einsetzen

347 Dienst mit Containern anwenden

426 Software mit agilen Methoden entwickeln

Suchen Sie sich ein Projekt aus, das zu Ihrem Modulablaufplan passt oder wählen Sie ein eigenes Projekt zur Repetition des Stoffes oder zum Kennenlernen von neuen Technologien aus. Besprechen Sie eigene Projekte mit der Lehrperson.

Hinweis: Es ist nicht erlaubt, Aufgaben für das Qualifikationsverfahren (LB, LBV, ...) während des Lernateliers zu realisieren. Das Lernatelier ist auch keine Aufgabenstunde.

Hinweis: Dimensionieren Sie Ihr Projekt so, dass Sie in der gegebenen Zeit fertig werden, aber auch genügend Arbeit vorhanden ist, um die Zeit zu füllen.

Ziele

Am Schluss dieses Lern- und Arbeitsauftrages sollen Sie ein fertiges Produkt mit Projektdokumentation erstellt haben.

Aufgaben

1) Informieren

a) Schliessen Sie sich in Gruppen zusammen, die mindestens aus zwei, maximal aus vier Personen bestehen.

b) Wählen Sie in der Gruppe ein Projekt aus. Beachten Sie den Schwierigkeitsgrad, Ihren Wissensstand und die zur Verfügung stehende Zeit bei der Auswahl. Sprechen Sie eigene Ideen mit der Lehrperson ab.

c) Machen Sie eine Anforderungsanalyse. Werden Sie sich in der Gruppe einig, was Sie genau realisieren möchten. Das Projekt sollte durch die Anforderungen so präzise wie möglich beschrieben sein. Legen Sie die minimalen Anforderungen und «nice-to-have»-Anforderungen fest, die bei genügend Zeit realisiert werden können.

d) Legen Sie fest, wo und wie Sie in der Gruppe Arbeitsergebnisse speichern und austauschen möchten.

e) Erarbeiten Sie sich noch fehlende Grundlagen für das Projekt.

Sie können die erarbeiteten Dokumente mit der Lehrperson besprechen, wenn Sie möchten.

Anforderung Nr.	Anforderung	Muss / Kann	Qualität / Funktional / Rand
1.	Transaktionen werden in Blöcken gespeichert.	Muss	Funktional
2.	Man kann neue Blöcke «minen».	Muss	Funktional
3.	Ein Block kann nicht abgeändert werden.	Muss	Funktional
4.	Man kann Transaktionen tätigen.	Muss	Funktional
5.	Man kann die Schwierigkeit zum Blöcke «minen» einstellen.	Muss	Funktional
6.	Die Blockchain beinhaltet ein Proof-Of-Work Verfahren.	Muss	Funktional
7.	Das Programm ist in NodeJS geschrieben	Muss	Rand

Quellen:

<https://dev.to/freakcdev297/creating-a-blockchain-in-60-lines-of-javascript-5fka>

<https://lisk.com/blog/posts/how-build-blockchain-javascript>

<https://github.com/Savjee/SavjeeCoin>

2) Planen

a) Teilen Sie das Projekt in klare, einzelne Aufgaben auf («Arbeitspakete»).

Nummer	Arbeitspaket	Dauer	Verantwortlicher	Deadline
1	Blockchain-Klasse erstellen.	2x45	Olivier Kistler	03.05.23
2	Block-Klasse erstellen	1x45	Olivier Kistler	03.05.23
3	Proof-Of-Work Verfahren	2x45	Olivier Kistler	03.05.23
4	Transaktions-Klasse erstellen	1x45	Olivier Kistler	10.05.23
5	Hash-Funktion erstellen	1x45	Olivier Kistler	10.05.23
6	Mining-Funktion erstellen	2x45	Olivier Kistler	10.05.23
7	Integrität der Blöcke implementieren	2x45	Olivier Kistler	17.05.23
8	Schwierigkeit für «mining» implementieren	1x45	Olivier Kistler	17.05.23
9	Webapplikation erstellen	3x45	Olivier Kistler	17.05.23

- b) Verteilen Sie diese Arbeiten an die Gruppenmitglieder und setzen Sie für jede Arbeit ein Abschlussdatum fest, dass mit dem Plan Ihrer Klasse übereinstimmt.
- c) Erstellen Sie eine ToDo-Liste mit verantwortlicher Person, Arbeitspaket und Abschlussdatum für das ganze Projekt.

Sie können diese ToDo-Liste mit der Lehrperson besprechen, wenn Sie möchten.

3) Entscheiden

Fällen und dokumentieren Sie wichtige Entscheidungen.

Ich habe mich dazu entschieden die Blockchain in NodeJS zu implementieren und ein Frontend mit HTML und CSS zu erstellen.

4) Realisieren

Erstellen Sie das Programm und die notwendigen Dokumente. **Erstellen Sie das Programm selbstständig und verwenden Sie nicht einfach das Resultat eines Lernvideos oder fertigen Code.**

5) Kontrolle

- a) Erstellen Sie Testfälle für das Programm.
- b) Testen Sie Ihr Programm und schreiben Sie ein kurzes Testprotokoll mit Testumgebung.

Sie können diese Testdokumente mit der Lehrperson besprechen, wenn Sie möchten.

Test-Nr.	Anforderungs-Nr.	Voraussetzung	Eingabe	Ausgabe
1.1	1	Server gestartet	1.Transaktion hinzufügen	2. Transaktion wurde hinzugefügt
2.1	2	Server gestartet, Transaktion hinzugefügt	1.Ausstehende Transaktionen minen	2. Der Block wurde gemined.
3.1	3	Server gestartet, Versuch Blockchain zu ändern	1.Transaktions amount ändern	2.Blockchain not valid
4.1	4	Server gestartet	1.Transaktion hinzufügen 3. Ausstehende Transaktionen minen	2.Hinzugefügt, nun ausstehende Transaktionen minen 4.Block gemined
5.1	5	Server gestartet, settings.html Page geöffnet	1.Schwierigkeit auf 3 ändern.	2. Schwierigkeit wurde auf 3 geändert
6.1	6	Server gestartet, Transaktion hinzugefügt	1.Transaktionen minen 3.So lange minen bis hash Schwierigkeitsstufe erreicht	2. mining wird gestartet 4. Block gemined

Test-Nr.	Testfall-Nr.	Datum	Tester	Resultat	Bemerkung	Unterschrift
1.1.1	1.1	24/05/2023	Olivier Kistler	OK	-	OK
2.1.1	2.1	24/05/2023	Olivier Kistler	OK		OK
3.1.1	3.1	24/05/2023	Olivier Kistler	OK	Siehe unten	OK
4.1.1	4.1	24/05/2023	Olivier Kistler	OK		OK
5.1.1	5.1	24/05/2023	Olivier Kistler	OK		OK
6.1.1	6.1	24/05/2023	Olivier Kistler	OK	Siehe unten	OK

Testfall 3:

Die Blockchain ist anfangs gültig aber sobald ich den Transaktionsbetrag ändere wird sie als ungültig erkannt.

```

32 console.log("Is chain valid? ", wampflerCoin.isChainValid());
33
34 wampflerCoin.chain[1].transactions[1].amount = 1000;
35 wampflerCoin.chain[1].hash = wampflerCoin.chain[1].calculateHash();
36
37 console.log("Is chain valid? ", wampflerCoin.isChainValid());
38 console.log(myWalletAddress);
39

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Starting miner
Block mined: 00a80a17c48ea180775c2fa3118c7c51dc74f347d2f2e64773627cc4ffcf75ab
Block mined!

Balance of Olivier is 270
Is chain valid? true
Is chain valid? false

```

Testfall 6:

Bevor ein Block tatsächlich gemined werden kann, muss zuerst der Hash solange neu kalkuliert werden bis er die von der Difficulty festgelegte Anzahl nullen am Anfang hat. In diesem Beispiel ist die Schwierigkeit auf 2 eingestellt.

```

Hash
00752d2d31b6b43d79e4462dd19d6d...

```

6) Auswertung

Überlegen Sie sich, was gut gelaufen ist und was eher nicht. Sie können diese Informationen in Ihrem Portfolioeintrag verwenden.

7) Portfolioeintrag

Schreiben Sie den Portfolioeintrag und reichen Sie ihn ein. Gehen Sie im Portfolioeintrag auf Ihre Ziele und deren Erreichung sowie die erstellte Dokumentation ein. Beschreiben Sie im Portfolioeintrag auch Ihren Lernweg, die Probleme, Erkenntnisse und Erfolge.

Projektvorschläge

165 NoSQL-Datenbanken einsetzen

Choose your own adventure

«Choose your own adventure» sind Bücher, in denen Ihnen eine Situation geschildert und verschiedene Entscheidungsmöglichkeiten gezeigt werden.

(<https://de.wikipedia.org/wiki/Spielbuch>). Erstellen Sie eine (Web-)Applikation, mit der Sie eine solche Geschichte durchspielen können. Speichern Sie alle Daten in einer NoSQL-Datenbank.

Repetitionshilfe

Erstellen Sie ein Programm, mit dem Sie Faktenwissen aus dem Unterricht repetieren können. Sie sollten Fragen und Antworten eingeben können und das Programm soll Sie abfragen.

Erweiterungen wie eine Statistik oder andere Fragetypen wie Multiple-Choice-Aufgaben sind möglich. Speichern Sie alle Daten in einer NoSQL Datenbank.

Medienmanager

Erstellen Sie ein Programm, in dem Sie Medien (Bilder, Video, Links, Audio, ...) speichern können. Dabei sollen alle möglichen Metadaten ausgelesen und in der Datenbank gespeichert werden können. Erstellen Sie eine Oberfläche, mit der Sie auf diese Metadaten zugreifen können.

Gallery-Maker

Erstellen Sie ein Programm, dem Sie eine Verzeichnishierarchie mit Bildern übergeben können und das Ihnen eine schöne Fotogalerie als statisches HTML ausgibt. Dabei soll es eine Übersicht mit Thumbnails und eine Detailansicht geben. Erweiterungsmöglichkeiten sind Texte, die zu den Bildern in einem geeigneten Format abgelegt werden können und das Auslesen der Metadaten.

347 Dienst mit Containern anwenden

Docker Visualisierung mit phaser.io

Mit phaser.io (<https://phaser.io/>) können Sie Spiele und visualisierungen für den Browser entwickeln.

Benutzen Sie die Ausgaben von «docker ps», «docker info», etc und stellen Sie Ihre Docker als 3D-Szene im Browser dar:

- Ein Docker kann ein Gebäude sein.
- Je mehr Platz/Memory er benötigt, desto grösser ist es.
- Jeder Prozess ist eine Spielfigur im Haus
- ...

Als Erweiterung könnten Sie beispielsweise Docker herunterfahren, indem Sie das Haus abbrechen (bitte mit Explosionen).

Dockercraft (schwieriges Projekt!)

Docker können innerhalb von Minecraft gesteuert werden: <https://github.com/docker/dockercraft>
Nehmen Sie dies als Basis für ein Projekt: Erweitern Sie Cuberite (den Minecraft Server) auf geeignete Art und Weise.

Achtung: Nur Minecraft Spielen ist kein Projekt!

426 Software mit agilen Methoden entwickeln

Implementieren Sie folgende Projekte mit den Design-Patterns im Hinterkopf. Versuchen Sie mindestens ein Pattern korrekt einzusetzen.

RPG-Games

Folgende Kurse beschreiben, wie man mit C# RPG's bauen kann. Nehmen Sie diese objektorientierten Programme als Grundlage und erweitern Sie diese:

- <https://roquesharp.wordpress.com/>
- <https://scottlilly.com/learn-c-by-building-a-simple-rpg-index/>

Textbasierte Multiuserwelt

Ein MUD (MultiUser Dungeon) ist eine textbasierte Welt, in der mehrere Benutzer gleichzeitig Abenteuer erleben, sich bekämpfen oder Aufgaben erfüllen können.

Installieren Sie <https://github.com/DavidRieman/WheelMUD>, erstellen Sie eine Welt und mit neuen Objekten.

Augmented Reality

Mit ARCORE (<https://developers.google.com/ar>) können Sie Augmented Reality Applikationen entwickeln. Arbeiten Sie sich ein und erstellen Sie ein eigenes, einfaches Beispiel.



Build your own x

Wollten Sie schon immer einmal ein eigenes Betriebssystem, einen eigenen Renderer, eine eigene Datenbank, einen eigenen Webserver oder eine eigene Blockchain entwickeln?

<https://github.com/codecrafters-io/build-your-own-x> hat Anleitungen dazu gesammelt.

Gütekriterien

Der Lern- und Arbeitsauftrag ist erfüllt, wenn ...

- Wenn Sie das Projekt nach IPERKA abgearbeitet haben.
- Wenn Sie eine lauffähige Applikation produziert haben.

Zusätzliche Angaben zum Auftrag

Keine.

Mögliche Erweiterungsaufträge

Keine.