

1996029

Wechseln zu: [Navigation](#), [Suche](#)

Die nachfolgenden Informationen wurden teilweise von [1926050](#) übernommen.

Nummer	1996029
Studiengang	Nachqualifikation IMS (NQ 2014)
Modul	IAM
Lehrveranstaltung	Multimedia-Anwendungen/Programmierung
Alternativname	Multimedia-Programmierung, MMProg
Kürzel	nq.iMM
Prüfer	Wolfgang Kowarschick
Zweitprüfer	Sabine Müllenbach
Prüfung	In diesem Semester wird eine Wiederholungspüfung angeboten.
Prüfungsart	Prüfungspraktikum, Studienarbeit
Details	Prüfungspraktikum: 2 h
Hilfsmittel	

Inhaltsverzeichnis

- [1 Termine](#)
- [2 Studienarbeit](#)
- [3 Wiederverwendung von Code](#)
- [4 Bewertung](#)

1 Termine

22. März 2020:

Abgabe der Anwendungsfälle (Use-Cases) via [Git-Repository](#) und Link in [Moodle](#)

29. März 2020:

Abgabe des Datenmodells: UML-Klassendiagramm (einschließlich textueller Beschreibung) sowie zugehöriger JavaScript-Dateien, die dieser Struktur genügen (die Implementierungen der Klassen und Funktionen dürfen noch fehlen)

Datenmodell als PDF-Datei sowie JavaScript-Dateien im [Git-Repository](#), Link auf den Ordner im Repository in [Moodle](#)

24. April 2020 (Uhrzeiten werden in Moodle bekanntgegeben):

Zwischenpräsentation eines **lauffähigen Prototyps** der Studienarbeiten (die Präsentation

eines Datenmodells reicht nicht aus)

5. Juni 2020 (Uhrzeiten werden in Moodle bekanntgegeben):

2. Zwischenpräsentation eines **lauffähigen Prototyps** der Studienarbeiten (es muss ein Fortschritt gegenüber der ersten Präsentation sichtbar sein)

Prüfungspraktikum entfällt

Die Teilnahme an der Prüfung ist ohne Prüfungspraktikum möglich.

12. Juli 2020

Abgabe der vollständigen Studienarbeit einschließlich einer [Erstellungserklärung](#) unter [Moodle](#) und als Inhalt eines Git-Repositories.

Juli 2020, J2.10 (die genauen Termine werden nach der Endabgabe bekanntgegeben):

Präsentation der Ergebnisse. Sie müssen Ihre Anwendung erklären und Ihre Anwendung demonstrieren können. Sie brauchen keine Präsentationsfolien vorzubereiten.

Beachten Sie, dass in dem Dokument [Erstellungserklärung](#) auch eine Abgabeliste enthalten ist, in der genau aufgelistet wird, was Sie in welcher Form abgeben müssen.

Darüber hinaus müssen Sie im Laufe des Semesters regelmäßig den aktuellen Stand Ihrer Arbeit im Git-Repository, das Ihnen am Anfang des Semesters zur Verfügung gestellt wurde, zwischenspeichern. Das heißt, vor jeder Zwischenabgabe müssen mindestens 5 Versionen der abzugebenden Dokumente ins Repository eingefügt werden. Von der eigentlichen Implementierung der Studienarbeit müssen vor der Endabgabe jeweils mindestens 30 substantiell verschiedene Versionen ins Repository eingefügt werden.

2 Studienarbeit

Als Teilnehmer der Lehrveranstaltung „Multimedia-Anwendungen/Programmierung“ (MMProg) sollen Sie nachweisen, dass Sie in der Lage sind, eine interaktive Web-Anwendung mit Hilfe von HTML 5 und JavaScript (ECMAScript 6) zu erstellen.

Sie müssen daher im Rahmen einer Studienarbeit eine derartige Anwendung gemäß den in der Vorlesung vermittelten Prinzipien realisieren. Am Ende des Semesters müssen Sie diese Arbeit präsentieren. Sie können sich das Thema (in Absprache mit dem Dozenten) selbst aussuchen.

Da sich die Vorlesung und das Praktikum an dem Thema Spiele orientieren, ist es sinnvoll ein Spiel als Studienarbeit zu programmieren.

Anbei finden Sie eine exemplarische und unvollständige Liste mit Themenvorschlägen:

Mini-Billard

Mini-Mohrhuhn

Geschicklichkeitsspiel: herabfallende Gegenstände sammeln, Ballerspiel, ...

Snake

Memory

Vier gewinnt, Fünf gewinnt, Reversi ... (zwei Spieler, evtl. auch ein Spieler + KI)

Point-and-Click (2D)

Jump-and-Run

Spielephysik

Tic Tac Toe (2-dimensional oder 3-dimensional)

Pong

Doodle Jump

Flappy Bird

Bubble Worlds

...

Beachten Sie, dass Sie im Falle einer Wiederholungsprüfung eine vollkommen neue Arbeit erstellen müssen.

3 Wiederverwendung von Code

Sie sind dazu berechtigt fremden Code in Ihrer eigenen Arbeit zu verwenden, solange dies durch dessen Lizenz erlaubt ist. Dazu zählen auch sämtliche Code-Beispiel und Musterlösungen aus der Lehrveranstaltung. Für jeglichen fremden Code müssen Sie die Quelle korrekt angeben. Ihre Studienarbeit kann als **mangelhaft** bewertet werden, wenn der Eigenanteil an der Arbeit zu gering ist, d. h., wenn Sie überwiegend vorgefertigten Code zur Realisierung der Arbeit einsetzen.

4 Bewertung

Sie müssen bei der Erstellung der Arbeit darauf achten, dass Sie verständlichen, lesbaren, modularen und wiederverwendbaren Code nach anerkannten objektorientierten Programmierprinzipien schreiben. Die Anwendung muss mit mit HTML5 und JavaScript erstellt werden.

In die Bewertung fließen ein:

Studienarbeit

- Angewendete Programmiertechniken
 - Klassen, Objekte, Objektbuilder
 - Event-Listener, Verarbeitung von Events
 - Modularisierung (Datei- und Code-Ebene)
 - Timer (setTimeout, setInterval)
- Qualität
 - Qualität der Modularisierung (Datei- und Code-Ebene)
 - Sinnvolle Benennung, Aussagekraft
 - Wiederverwendbarkeit, Konfigurierbarkeit
 - Korrektheit, keine Fehler
 - Keine Redundanz (DRY)
 - Logic-Data-View-Controller(-Service)-Prinzip
 - Qualität der Model-Module
 - Qualität der View-Module
 - Qualität der Controller-Module
 - Qualität der Logik-Module
- Tiefe
 - Komplexität des selbst erstellten Codes
- Dokumentation (Use-Cases, Klassendiagramm samt Beschreibungen)
 - Vollständigkeit
 - Verständlichkeit
 - Korrektheit
 - Inline-Kommentare (im Sourcecode)
 - Quellenangaben

Prüfungspraktikum

- Qualität und Korrektheit der Lösungen

Die Bewertung **einer der Hauptkategorien** „Programmiertechniken“, „Qualität“, „Umfang“, „Dokumentation“, „Prüfungspraktikum“ als **mangelhaft** führt dazu, dass die **gesamte Arbeit** als **mangelhaft bewertet** wird. Eine Arbeit, die nicht gemäß einem Pattern wie MVC oder LDVC strukturiert ist, wird höchstens als *ausreichend* bewertet.

Die gesamte Arbeit wird als **mangelhaft** bewertet, sobald auch nur ein **Plagiat** enthalten ist.

Die Arbeit wird als unvollständig und damit als **nicht abgegeben** gewertet, wenn **eine der Zwischenabgaben nicht erfolgt** ist.

Die Arbeit wird als unvollständig und damit als **nicht abgegeben** gewertet, wenn ein wesentlicher Bestandteil fehlt, d.h., wenn **in der Abgabeliste** im Anschluss an die Erstellungserklärung **eine wesentliche Frage mit nein beantwortet wurde oder wenn die Erstellungserklärung samt Abgabeliste fehlt**.

Die Arbeit wird als unvollständig und damit als **nicht abgegeben** gewertet, wenn **die grobe Entstehungsgeschichte der Arbeit nicht anhand der Besprechungen im Praktikum oder anhand der Versionen im SVN-Repository nachvollzogen werden kann** (entweder **regelmäßige** Teilnahme am Praktikum oder mindestens 5 einschlägige Updates vor jeder Zwischenabgabe und anschließend mindestens 20 einschlägige Updates vor der Endabgabe).

Kategorie:
Prüfungsnummer

Diese Seite wurde zuletzt am 11. November 2014 um 14:08 Uhr bearbeitet.
Inhalt verfügbar unter [CC BY-SA 4.0](#).

