

Thema:	Abschlussaufgabe der Veranstaltung „Objektorientierte Analyse und Design“
Autor:	Prof. Dr. Stephan Kleuker
Version/Datum:	1.0./ 2.3.2022

Die folgende Aufgabe soll in einem Team von *drei* Studierenden bearbeitet werden. Andere Gruppengrößen sind mit dem Veranstalter abzusprechen. Der organisatorische Hintergrund der Veranstaltung ist dem Modulhandbuch des Studiengangs zu entnehmen.

Hinweise bei Wiederholung

Sollten Sie ein OOAD-Praktikum bereits bestanden und die Prüfung versäumt haben bzw. durchgefallen sein, sind Sie dazu verpflichtet sich aktiv um eine Arbeitsgruppe zu kümmern. Falls Sie keine Personen finden, melden Sie sich dann spätestens drei Wochen vor Vorlesungsende bei mir, Sie werden dann entweder einer Zweiergruppe oder einer Gruppe von wiederholenden Personen zugeordnet.

Aufgabenstellung

Suchen Sie sich oder entwickeln Sie ein Framework oder eine Bibliothek, das Vorgaben zur sinnvollen Nutzung (z. B. in der Art von anzuwendenden Design Pattern oder eines Frameworks) macht und entwickeln Sie ein kleines prototypisches Projekt mit den bekannten OOAD-Mitteln, in dem Sie diese Teilfunktionalität anhand eines implementierten Beispiels vorstellen. Beispiele für Frameworks können sein:

- Die Kommunikation zwischen GUI-Elementen und GUI und Business-Klassen in einem GUI-Framework, wie sie z. B. in Java mit Swing oder JavaFX (OpenJFX) machbar ist; Sie entwickeln dazu ein kleines Beispiel mit verschiedenen GUI-Elementen, deren Zusammenarbeit sie zeigen wollen
- Nutzung eines objekt-relationalen Mappers mit dem Objekte einfach in relationalen Datenbanken gespeichert werden können, wie es z. B. in Java durch Realisierungen von JPA oder Ebean möglich ist
- die Verbindung zwischen Web-Oberfläche und Geschäftslogik, wie sie z. B. mit dem Google Web Toolkit möglich ist
- Erstellung und Verknüpfungsmöglichkeiten von WebServices
- Die Nutzung einer Infrastruktur zum Austausch von Informationen, wie z. B. MQTT, aber auch WebSockets oder Server-Sent Events
- Möglichkeiten Java-Byte-Code zu bearbeiten, z. B. durch Javassist
- Nutzung von Metaklassenprogrammierung, die z. B. auf der Java-Klasse Class basiert und z. B. zur Analyse und Ausführung von Annotationen genutzt wird; Sie könnten ein eigenes minimales auf Annotationen basierendes Framework schreiben
- Entwicklung einer Web-Applikation mit Teilen des Frameworks Java-Spring
- Analyse der Umsetzungsmöglichkeiten ausgewählter objektorientierter Design-Pattern (z. B. Observer-Observable, Decorator) in nicht klassisch objekt-orientierten Sprachen wie JavaScript, TypeScript, Go, Python ...

Wenn Java in der obigen Liste genannt wurde, sind auch andere objekt-orientierte Sprachen denkbar.

Die Themenliste ist absolut unvollständig, sprechen Sie eigene Ideen mit dem Dozenten spätestens in der vorletzten Veranstaltungswoche ab. Das gewählte Thema darf nicht Schwerpunkt einer bisherigen oder schon geplanten Prüfungsleistung sein.

Vorschlag zum Finden eines Themas:

1. Suchen Sie nach für Sie interessante Technologien, probieren Sie diese aus, prüfen Sie, ob Sie deren Ansatz verstehen und wählen Sie dann eine Technik aus. Dies ist in den ersten Veranstaltungswochen möglich.
2. Formulieren Sie sich selbst eine Aufgabenstellung mit der Sie die gewählte Technologie vorstellen wollen. Denken Sie beim Umfang z. B. an ein Übungsblatt mit einer zentralen Aufgabe, mit der die Technik verständlicher werden kann. Alternativ planen Sie eine größere Aufgabe, von der Sie ein erstes, für ihre Technik spannendes, Inkrement realisieren.
3. Führen Sie dann eine vollständige Entwicklung zur Lösung ihrer Aufgabenstellung **mit den aus OOAD am Ende der Veranstaltung bekannten Mitteln** durch. Sie können dabei weniger relevante Details, z. B. das Design und die Ergonomie bei einer GUI-Technik oder unwesentliche Objektvariablen von Klassen, vernachlässigen.
4. Bedenken Sie, dass Ihr resultierendes Programm (nur) ein wichtiges Teil des Ergebnisses ist, d. h. der Umfang der Implementierung steht nicht im Mittelpunkt, es geht um „AD“ und deren Zusammenhang.

Ihre Ergebnisse werden in einer Hausarbeit, einer einfachen schriftlichen Arbeit, festgehalten. Der typische Aufbau sollte wie folgt aussehen, Abweichungen und Ergänzungen können aufgabenindividuell natürlich sinnvoll sein (wie sie z. B. in der Word-Beispieldokumentenstruktur angedeutet wird).

1. Einleitung: worum geht es grob in der betrachteten Technologie (Motivation), prägnante, kompakte Formulierung der Aufgabenstellung, Überblick über die Hausarbeit (zusammen max. eine Seite)
2. Grundlagen: Vorstellung der Technologie; dies kann hier bereits sehr detailliert mit Beispielen erfolgen, was dann das 5. Kapitel kürzer macht; sollte SW installiert werden müssen, die über die vorgegebene SEU hinaus geht, ist die Installation in einem Anhang zu dokumentieren
3. Anforderungsanalyse: führen Sie eine Analyse mit dem in der Veranstaltung vorgestellten Ansatz durch, als Ausgangspunkt können ein- oder mehrere Use Cases dienen (abhängig von der Aufgabenstellung, reicht eventuell ein Use Case aus)
4. Modellierung: Erstellen Sie ein Klassendiagramm, das auch in der Implementierung genutzt wird; erklären Sie kurz die genutzten Klassen und Ideen dahinter; abhängig von der Aufgabenstellung können weitere Visualisierungen sinnvoll sein
5. Implementierung: Stellen Sie kurz wichtige Code-Fragmente, nicht den gesamten Code vor; gehen Sie darauf ein, wie Sie die Erfüllung der Anforderungen überprüft haben
6. Fazit: schreiben Sie ihre persönlichen Erfahrungen auf; auch gescheiterte Ansätze sollen genannt werden, da sie wichtige experimentelle Ergebnisse sind
7. Anhänge: optional, z. B. sinnvoll für Installationsanleitungen

- Literaturverzeichnis: verwendete Quellen; Bücher und Fachtexte sind besser als Links, die natürlich auch notwendige Quellen sein können; das Literaturverzeichnis hat keine Kapitelnummer

Verwenden Sie für den Bericht als Schrifttyp Times New Roman oder einen ähnlichen Typen mit Serifen mit Schriftgröße 12 pt und 15 pt Zeilenabstand. Sie können die Vorlage von der Veranstaltungswebseite nutzen. Halten Sie sich an einen akademischen Schreibstil, vermeiden Sie z. B. Worte wie „ich“ und „wir“, außer wenn Sie am Ende des Berichts ein persönliches Fazit ziehen sollten. Lesen Sie sich vor der Erstellung der Hausarbeit auch <http://kleukey.iui.hs-osnabrueck.de/querschnittlich/AnforderungenAbschlussarbeit.pdf> durch.

Zu liefernde Produkte

Abzugeben in elektronischer Form per einfach zu nutzendem NetCase-Link ist ein zip-File dessen Namen aus den Nachnamen der Personen der Gruppe besteht.



↓ Download Nase1_Nase2_Nase3.zip (147.2 MB)

Direkter Link <https://netcase.hs-osnabrueck.de/index.php> download

Inhalte sind:

- bei Java-Projekten: der Quellcode des Projekts als *Eclipse-Java-Projekt* in gepackter Form (.7z- oder .zip-Format). Der Code ist mit JavaDoc zu kommentieren und muss Java-Konventionen in der Benennung von Variablen und Nutzung von Methoden folgen. Eine weitere Kommentierung innerhalb der Methoden sollte durch die Dokumentation der Methoden, die Verwendung „sprechender“ Variablennamen und die Kürze der Methoden nicht notwendig sein. Ihr Projektname besteht aus den Nachnamen der Projektteilnehmer. Das Projekt muss in der zur Verfügung gestellten Entwicklungsumgebung (KleukeySEU) lauffähig sein.
bei anderen Sprachen oder Java Enterprise Versionen: Projektdateien mit Nennung der genutzten freien Entwicklungsumgebung und dokumentierter Code. Das Projekt muss unter Windows 10 durch Ihre Installationsanleitung lauffähig werden.
- die eigentliche Hausarbeit, zumindest in PDF-Form. Die Arbeit soll mindestens 12 fachliche

Seiten umfassen, längere sinnvolle Texte sind erlaubt, es gibt formal keine Obergrenze. Es ist sicherlich möglich auch in zwölf Seiten ein sehr gutes Ergebnis zu erreichen. Codefragmente können im Text vorkommen. Ergänzen Sie eine Lizenz für Ihre Aufgabenstellung und die Software, z. B. MIT oder Apache License.

- Stundenzettel. Eine grobe Übersicht, an welchem Aufgabenteil Sie wie lange gearbeitet haben. Der Zettel wird für statistische Zwecke benötigt und geht nicht in die Bewertung ein, da hier *nur die Arbeit und nicht die Leistung bewertet wird*.
- Übersicht über Hauptverantwortliche für Teilaufgaben bei der Erstellung. Geben Sie für alle Teile der Arbeit an, wer sie erstellt hat; bei einer gemeinsamen Bearbeitung ist immer ein Hauptverantwortlicher zu benennen. Die Bewertung erfolgt als Individualnote basierend auf dem Gesamteindruck der Arbeit. Teile ohne Nennung eines Hauptverantwortlichen fallen aus der Bewertung.

Weiterhin muss eine kurze Erklärung in das PDF eingebunden werden, dass die Aufgabe selbstständig unter ausschließlicher Nutzung der in der Dokumentation genannten Hilfsmittel bearbeitet wurde.

Prüfen Sie vor der Abgabe z. B. mit einer neuen leeren Eclipse-Installation der KleukerSEU auf einem anderen Rechner, ob Ihre Software problemlos lauffähig ist.

Abgabe/Abnahme

Die elektronische Abgabe muss bis spätestens Mo 8.8.2022, 23:59 Uhr, erfolgen, spätere Abgaben sind *nicht* möglich. In Ausnahmefällen werden Sie per E-Mail aufgefordert ihr Projekt persönlich vorzustellen. Notwendige Termine werden individuell vereinbart. *Die Bewertung findet auf Basis der Dokumentation und der gelieferten Software statt.* Die Ergebnisse werden Ihnen in einer E-Mail an alle Gruppenteilnehmer mitgeteilt.

Da am Ende des regulären vierten Semesters einige Projektarbeiten anstehen können, wird dringend dazu geraten, die Aufgabe frühzeitig zu beginnen. Dies ist bereits sinnvoll mit der Nennung der Aufgabenstellung möglich.

Aufdatierung der Informationen

Ergänzende Informationen zur Aufgabe, Organisation und Bewertung können ggfls. der Veranstaltungsw Webseite entnommen werden. Die regelmäßige Kontrolle dieser Seite maximal im Wochenabstand auf mögliche Veränderungen ist verpflichtende Aufgabe aller Teilnehmenden.

Abschlussbemerkung

Abschließend sei daran erinnert, dass sehr gute Leistungen nicht durch reine Pflichterfüllung entstehen. Eine mit „befriedigend“ zu bewertende Arbeit wird in der Prüfungsordnung als „eine Leistung, die in jeder Hinsicht durchschnittlichen Anforderungen entspricht“ beschrieben.