

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

WICHTIG

Die Online-Befragung zur genutzten alternativen Veranstaltungsform und zur Lehrevaluation ist online. Bitte ausfüllen: <https://forms.gle/NxVd8uXszQgxvvXk9>. Sie werden eventuell aufgefordert sich bei Google anzumelden, das ist nur notwendig, wenn Sie in der Bearbeitung eine Pause machen wollen und das Teilergebnis zwischenspeichern wollen. Die Befragung endet am 5.6, die Ergebnisse stehen im nachfolgenden Fragen&Antworten-Dokument auf der Webseite der Veranstaltung.

Frage: Wo ist eigentlich der Unterschied zwischen dem Strategy Pattern und „normaler“ Vererbung?

Antwort: (Fast) jedes Pattern basiert auf dynamischer Polymorphie, also damit auch Vererbung. Völlig richtig, ist dies beim Strategy Pattern, dem eventuell einfachsten Pattern nicht sonderlich erkennbar. Dabei ist die Basisidee, einen Algorithmus zu kapseln und so als eine Art Funktionsobjekt austauschbar zu machen sehr interessant und sehr wichtig. Ein Anwendungsbereich in Java sind die Functional Interfaces, die u. a. in Lambda-Ausdrücken genutzt werden, die diesen Ansatz nutzen. In anderen Programmiersprachen sind Funktionen normale Elemente, die einfach weitergegeben und zugewiesen werden können, wie es z. B. in C, JavaScript und Go der Fall ist.

In der Praktikumsaufgabe wird der echte Strategy-Ansatz, hier mehrere Methoden austauschbar zu machen, mit der klassischen Vererbung vermischt. Eine Motivation ist, dass Pattern oft nicht in der Reinform vorkommen. Die Praktikumsaufgabe ist sicherlich die einfachste der größeren Aufgaben zu Pattern. Sie macht aber zumindest einigen Studis Spaß und andere Studis, die Probleme mit dieser sehr einfachen Aufgabe hatten, wissen, dass Sie in ihrer akademischen Entwicklung noch einen langen Weg zum Bachelor haben. Das Software-Engineering-Projekt kann zum Aufholen geringer Defizite genutzt werden.

Frage: Ich wollte einmal Fragen bis wann wir die Hausarbeitsthemen nennen sollen.

Antwort: Spätestens in der letzten Vorlesung in meiner Online-Sprechstunde. Es geht auch per Mail: Wer mit wem, fachliche Idee, eingesetzte Technik, Beispiel:

Nase X, Nase Y, Nase Z, Geburtstagswunschkiste von Kindern, Nutzer können Eintragen was gewünscht ist und wer was kauft (nur Kernfunktionalität ohne Nutzerauthorisierung, ...), JavaFX mit DB im Hintergrund (Verbindung z. B. hart in SW des Clients reingecoded oder Konfig-Textdatei ohne detaillierte Prüfung)

Vom Umfang kalkuliere ich, dass gute Studierende jeweils ca. 40-60 Stunden in die Arbeit hineinstecken, bei sehr guten kann es weniger sein, eine obere Grenze bei unerfahreneren Leuten gibt es leider nicht. Denken Sie daran, dass das A von OOAD auch wichtiger Teil der Aufgabe ist.

Frage: Viele der Design-Pattern und OOAD-Ideen stecken in Bibliotheken oder Frameworks drin, die wir in der Hausarbeit nutzen wollen, ist das ein Problem?

Nein, sie werden in der Hausarbeit die genutzten Technologien nennen und auf die implizit enthaltenen Pattern in ihrer Dokumentation, z.B. den Grundlagen, hinweisen. Es kann dadurch passieren, dass Sie kein Pattern selbst programmieren.

Frage: (aus Diskussion) dabei wird auch das Singleton-Pattern eine große Rolle spielen...

Antwort: Das Singleton-Pattern ist eher ein Anti-Pattern, da es dem OO-Prinzip, eine Klasse steht für die Erzeugung beliebig vieler Objekte, widerspricht. Das Prinzip trifft natürlich nicht auf alle Klassen zu, z. B. Controller-Klassen, aber generell sollte die Möglichkeit der Existenz mehrerer Objekte, z. B. bei verteilten Systemen, in Betracht gezogen werden. Das heißt nicht, dass Sie das Singleton-Pattern nicht nutzen sollen, da es Stellen gibt an denen es sinnvoll und notwendig ist. Sie sollen nur nicht ihre Architektur auf Singletons ausrichten.

Frage: Wie detailliert müssen wir unser Programm in der Ausarbeitung beschreiben

Antwort: Schwierig zu verallgemeinern, aber die Prüfung Ihrer Programmierfähigkeiten steht nicht im Mittelpunkt der Hausarbeit. Diese wurden bereits geprüft, Defizite führen aber zu deutlich schlechteren Bewertungen. Da Sie Ihre Modellierung und den Weg dahin detailliert beschreiben und die Programmierung nur noch die Umsetzbarkeit zeigt, ist das Kapitel zur Implementierung eher kurz. Es können z. B. ein bis n typische Abläufe mit Blick auf die Umsetzung beschrieben werden. Code-Schnipsel für wichtige Stellen sind natürlich erlaubt. Eine „zu detaillierte“ Beschreibung der Implementierung führt nicht zu Abzügen in der Note, kann diese aber auch nicht verbessern. Nur der Ansatz „Wir schreiben 20 Seiten über die Implementierung, da wir uns da auskennen und den Rest fassen wir auf 3 Seiten zusammen, da wir nicht wissen was wir schreiben sollen.“ kann nicht zu einer brauchbaren Note führen.

Frage: Können wir auch Maven-Projekte abgeben?

Antwort: Generell ja, beachten Sie, dass Sie dann kurz den Maven-Ansatz erklären sollten, was z. B. in einem Anhang mit einer Erklärung Ihrer pom.xml und der Schritte zum ausführbaren Programm geschehen kann.

Generell laufen Maven-Projekte in der KleukerSEU, es kann aber zu einem unfreundlichen Fehler kommen, wenn ein neues Maven-Projekt erstellt wird und Sie die UMLet-Erweiterung von Eclipse selbst installiert haben. Die Erweiterung wäre dann erst zu deinstallieren.