Hausarbeit im Modul Softwaredesign

„Lagerverwaltung“

Marius Mamsch
Gescherweg 165
48161 Münster

Niklas Devenish
Töpferstraße 91
48165 Münster

Jonas Elfering
Hagen 3
48624 Schöppingen

Hochschule Weserbergland
Studiengang: Wirtschaftsinformatik
Studiengruppe: WI44/14
Betreuender Dozent: Prof. Dr. Robert Mertens

Ausbildungsbetrieb:
Fiducia und GAD IT AG
GAD-Straße 2-6
48163 Münster

Sperrvermerk

Die vorliegende Arbeit beinhaltet Informationen über betriebsbezogene Prozesse sowie vertrauliche Daten des Unternehmens.

Sie darf nicht ohne ausdrückliche Genehmigung der

GAD eG

GAD-Straße 2-6

48163 Münster

veröffentlicht oder Dritten zugänglich gemacht werden.

I Inhaltsverzeichnis

II Abkürzungsverzeichnis

Abkürzung	Beschreibung

IV Abbildungsverzeichnis

V Tabellenverzeichnis

1 Verwendete Patterns

1.1 Model-View-Controller

1.2 Observable/Oberserver

In der GUI wurde das Observable/Observer Pattern genutzt um Veränderungen in den Models angemessen in der GUI darzustellen. Das LagerVerwaltungsModel implementiert das Interface Observable und wird von der LagerVerwaltungsView, die den Observer erweitert, observt. Das dient vor allem, dazu Änderungen an der laufenden Buchung im LagerVerwaltungsModell zu erkennen und die GUI hinsichtlich dieser Änderungen zu aktualisieren. Aber auch auf Änderungen der Struktur der Lager und auf Änderungen in der Buchungsliste wird hier geachtet, damit diese immer korrekt angezeigt werden.

Desweiteren werden Observable/Observer bei den LagerModels, die das Interface Observable implementieren, eingesetzt. So erweitert der LagerBaumKnoten den Observer und jeder Lnoten überwacht ein Lager, welches er in der Baumstruktur repräsentiert. Hierdurch wird es ermöglicht, dass die Knoten immer den aktuellen Bestand des Lagers anzeigen, ohne immer den kompletten Baum aktualisieren zu müssen. Außerdem überwacht die DetailView als Observer das LagerModel, welches momentan angezeigt wird. Auch wird dadurch ermöglicht, dass die DetailView die aktuellen Informationen zum Lager anzeigt ohne ständig neu erstellt oder komplett aktualisiert zu werden.

Eine weitere Observable/Observer-Beziehung ist zwischen dem Controller und der BuchungsBar zu finden. Der Controller ist in diesem Fall das Observable und die BuchungsBar der Observer. In diesem Fall teilt der Controller seinem Observer mit, wenn sich der Redo- oder der Undo-Stack geändert haben. Aufgrund von diesen Informationen entscheidet die BuchungsBar, ob die Button für den Redo-, bzw. Undo-Mechanismus aktiviert oder deaktiviert werden.

1.3 Command

1.4 Strategy

Das Entwurfsmuster Strategy wurde im Kontext des Sortierers eingesetzt. Der Sortier besitzt eine Sortierstrategie. Die Sortierstrategie ist ein Interface, welches die Methode sortiere() bereitstellt. Dieser Methode müssen zwei BuchungsModel mitgegeben

werden, die verglichen werden sollen. Die Methode sortiere() liefert true zurück wenn das zuerst mitgegebene BuchungsModel vor dem zweiten BuchungsModel angezeigt werden muss, ansonsten false. Es gibt mehrere konkrete Implementationen dieses Interface. Für jede mögliche Art die Buchungsliste zu sortieren eine. Diese konkreten Implementationen unterscheiden sich vor allem durch das Attribut nach welchem sie sortieren und durch die Sortierreihenfolge, beides ist am Namen der Implementationen erkennbar. Das Entwurfsmuster Strategy bietet den Vorteil, dass man nur einen Sortieralgorithmus implementieren muss, der mit dem Interface Sortierstrategie arbeitet, um alle Sortiermöglichkeiten abzubilden. Denn die konkrete Implementation der Sortierstrategie kann noch zur Laufzeit geändert werden. Diese Implementation ist sehr viel eleganter, als alternativ für jede Möglichkeit des Sortierens einen kompletten Sortieralgorithmus zu implementieren, denn das würde redundanten Code erzeugen.

2 Besonderheiten

3 Prototypen und Mockups

4 Zusammenfassung

5 Quellenverzeichnis

VI Anhangsverzeichnis

VIIAnhang

A1Abbildungen

Tabelle 1: Tabelle Anhang

A1.1 Eigene Grafiken

A1.2 Bilder

Abbildung 1: Blumen

A1.2.1 Aus Internet

A1.2.2 Aus Büchern

A2Glossar

Eigenständigkeitserklärung

„Wir versichern hiermit, dass wir diese Arbeit selbstständig verfasst, keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht haben. Das gleiche gilt auch für eingefügte Zeichnungen und, Kartenskizzen und Darstellungen."

Hameln, den 17.12.2015

Ort, Datum	Unterschrift

Hameln, den 17.12.2015

Ort, Datum	Unterschrift

Hameln, den 17.12.2015

Ort, Datum	Unterschrift