

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 1

Seite 1 von 2

In der Vorlesung haben Sie das Architekturmuster MVC kennengelernt. Dieses eignet sich gut für die Implementierung von interaktiven Softwaresystemen mit Benutzeroberflächen. Dabei wird das System in die drei Einheiten (Model, View und Controller) unterteilt. Diese Aufteilung bringt unter anderem den Vorteil der Wiederverwendbarkeit von Komponenten.

Während der Bearbeitung der Übungen soll ein PC-HardwareShop erstellt werden. Jedes Übungsblatt beschäftigt sich mit einem Teil des Shops. Achten Sie also darauf, dass Sie wiederverwendbare Komponenten schreiben.

Aufgabe 1

Model, View und Controller

Zunächst soll das Grundgerüst des Programms mithilfe des MVC-Architekturmusters implementiert werden.

- a) Erzeugen Sie die Klassen *Product*, *ProductList* und *Order*, die die gleichnamigen Interfaces aus dem Paket *fpt.com* implementieren.
Die Klasse *Product* soll als Repräsentation eines Produkts dienen und beinhaltet den Namen, die ID, den Preis und die vorhandene Stückzahl eines Produkts.
Die Klasse *Order* enthält zu den Waren auch die bestellte Anzahl und deren Preis.
Bei den Klassen *Order*, *ProductList* kann es sich um eine Ableitung der Klasse *java.util.ArrayList* handeln, die ausschließlich Waren enthält.
- b) Schreiben Sie die Klasse *ModelShop*, welche von der Klasse *java.util.Observable* erbt und *fpt.com.ProductList* implementiert und alle Methodenaufrufe an ihre Implementierung der *ProductList* weiterreicht.
Die Klasse *ModelShop* ist für die Verwaltung der Daten zuständig. Werden dem Model über den Controller Eingaben des Benutzers mitgeteilt, muss es sich aktualisieren und alle Views über die Änderung informieren (Observer-Pattern). Ein Beispiel ist der Vorlesung zu entnehmen.
- c) Implementieren Sie die Klasse *ViewShop*, welche das Interface *java.util.Observer* implementiert.
Die Klasse *ViewShop* dient als Benutzerschnittstelle für den Shop.
Es sollen die eingetragenen Produkte angezeigt werden.
- d) Schreiben Sie die Klasse *ControllerShop*, diese soll für die Verarbeitung der Benutzereingaben des Händlers zuständig sein.
Wird ein neues *Product* hinzugefügt, müssen diese Änderungen über das ControllerShop-Objekt dem Model mitgeteilt werden.
- e) Es soll möglich sein neue Produkte einzufügen und alte zu löschen. Erweitern Sie ihre Klasse *ViewShop* um genügend GUI-Elemente.
Implementieren Sie die nötige Funktionalität nach dem MVC-Prinzip.
Prüfen Sie die zu machenden Eingaben so früh wie möglich, um Fehler zur Laufzeit zu vermeiden.
Defekte Programme bzw. Fehler bei der Abnahme führen zu Punktabzug.

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 1

Seite 2 von 2

ViewShop 0.1

Apfel
Price:1.0
Quantity:1

Name
Apfel

Price
1

Quantity
1

add()

deleteSelected()

Abbildung 1: Beispiel Produkt einfügen

Aufgabe 2

Bestellformular ViewCustomer

Schreiben Sie hier nur die grafische Oberfläche ohne Funktionalität.

(Die Implementierung des *ControllerCustomer* wird in späteren Aufgaben folgen.)

- Erstellen Sie eine Kundenansicht *ViewCustomer*, welche das Interface *java.util.Observer* implementiert. Die Klasse *ViewCustomer* soll alle Produkte so anzeigen, dass der Name, der Preis und dessen Anzahl sichtbar ist. Außerdem soll die Möglichkeit bestehen, eine Stückzahl, die bestellt werden soll, zu wählen.
- Die Anzeige soll zudem eine Historie der getätigten Bestellungen anzeigen.

Order with 33 Items for 1333,62
31 x test1 (1271,00), 2 x test2 (62,62)

Name	Price	MaxCount	OrderCount
Apfel	1,00	1	

buy()

Abbildung 2: Bestellanzeige

HINWEIS: Ihr Programm muss nur die beschriebene Funktionalität erfüllen. Die graphischen Nutzeroberflächen in Abbildung 1 dienen nur als Beispiele. Sie müssen nicht im Detail nachgebaut werden.