Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

Entwicklung komplexer Software-Systeme

Praktikumsblatt 1

Ziel: Implementierung von Entwurfsmustern

Abgabe der ausgedruckten Lösungen: Freitag, 11.11., 13 Uhr im Postkasten im Informatik-Labor (ZW-7-17)

Achtung: Seien Sie sorgfältig bei der Implementierung. Es ist sehr wichtig, dass Sie die Aufgabenstellung exakt und korrekt umsetzen. Verwenden Sie keine öffentlichen Attribute.

Aufgaben:

H1.1 Entwurfsmuster „Abstrakte Fabrik“ implementieren

Erstellen Sie eine „Abstrakte Fabrik“ für den folgenden Anwendungsfall:
In unserem Internet-Versandhandel sollen Geschenkpakete konfiguriert werden können. In unserem Framework sehen wir deshalb eine abstrakte Klasse „Geschenkpaket“ vor: Ein Geschenkpaket besteht aus einem Werbegeschenk und einem Gutschein.
Für unseren Pilot-Anwender sehen wir die folgenden konkreten Geschenkpakete vor:

- Weihnachtspaket, bestehend aus einen Weihnachtsmann und einem CD-Gutschein mit dem Wert 20 €
- Osterpaket, bestehend aus einem Osterhasen und einem CD-Gutschein mit dem Wert 15€
- Familienpaket, bestehend aus einem Fußball und einem DVD-Gutschein mit dem Wert 25€.

Weitere Informationen zu den Klassen:

- Es gibt zwei konkrete Gutscheine: für eine CD und eine DVD. Ein Gutschein besitzt ein Attribut „wert“ vom Typ `Integer`, welcher den Euro-Wert darstellt.
- Es gibt drei konkrete Werbegeschenke: Weihnachtsmann, Osterhase und Fußball. Diese können Sie als eigene Klassen realisieren. Diese besitzen ein Attribut „art“ vom Typ `String`, welches die Art des Werbegeschenks nochmals als String enthält.

Ihre Aufgaben:

a) Erstellen Sie die erforderlichen Klassen mit allen notwendigen Operationen und Attributen entsprechend dem Entwurfsmuster „Abstrakte Fabrik“. Verwenden Sie für jede Klasse eine eigene Datei.

Beachten Sie: Die Lösung muss gewährleisten, dass der Klient bei der Zusammenstellung der verschiedenen Pakete nicht wissen muss, um welches Werbegeschenk und welchen Gutschein es sich konkret handelt!
Weiterhin muss sichergestellt sein, dass der Administrator den Wert der Gutscheine jedes Pakets nur an einer einzigen Stelle ändern muss.

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

Verwenden Sie die folgende Klasse `Klient` zum Testen Ihrer Lösung.

```
public class Klient {
  public static void main(String[] args) {
        Geschenkpaket weihnachten = new Weihnachtspaket();
        Gutschein meinGutschein = weihnachten.erzeugeGutschein();
        Werbegeschenk meinGeschenk =
weihnachten.erzeugeWerbegeschenk();
        System.out.println("Weihnachtspaket: "+ meinGutschein.getWert
+ " " + meinGeschenk.getArt);
        Geschenkpaket ostern = new Osterpaket();
        Gutschein meinGutschein2 = ostern.erzeugeGutschein();
        Werbegeschenk meinGeschenk2 = ostern.erzeugeWerbegeschenk();
        System.out.println("Osterpaket: "+ meinGutschein2.getWert + "
" + meinGeschenk2.getArt);
        Geschenkpaket familie = new Familienpaket();
        Gutschein meinGutschein3 = familie.erzeugeGutschein();
        Werbegeschenk meinGeschenk3 = familie.erzeugeWerbegeschenk();
        System.out.println("Familienpaket: "+ meinGutschein3.getWert +
" " + meinGeschenk3.getArt);
} }
```

H1.2 Entwurfsmuster „Beobachter“ implementieren

Für die Einkaufsliste sollen die Beobachter "WebDarstellung" und "Rechnung" realisiert werden.

a) Implementieren Sie zunächst die (unbeobachtete) Einkaufsliste aus der Vorlesung mit den Operationen

- `addArtikel(a:Artikel)`: fügt Artikel a zur Einkaufsliste hinzu
- `entferneArtikel(a:Artikel)`: entfernt Artikel a aus der Einkaufsliste

und die Klasse Artikel aus der Vorlesung mit den Spezialisierungen CD, DVD und Buch.

b) Erstellen Sie die abstrakten Klassen des Beobachter-Musters mit allen erforderlichen Eigenschaften.

<u>Beachten Sie:</u> Die Implementierung des allgemeinen Teils des Beobachter-Musters muss derart gestaltet werden, dass damit auch andere Objekte beobachtet werden könnten. Es muss also allgemein und abstrakt realisiert sein.
Sie können aber zusätzlich zu der parameterlosen aktualisiere-Methode eine weitere aktualisiere-Methode mit beliebigen Parametern einführen.

c) Erweitern Sie nun die Klasse `Einkaufsliste` um die Eigenschaften eines Subjektes im Sinne des Beobachter-Musters und führen Sie die folgenden Beobachter als Klassen ein:

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

- `WebDarstellung`: druckt bei einer Zustandsänderung der Einkaufsliste den hinzu gefügten bzw. entfernten Artikel auf der Konsole aus. Überlegen Sie sich einen Mechanismus, so dass ausgegeben werden kann, ob der Artikel hinzugefügt oder entfernt wurde.
- `Rechnung`: berechnet und druckt bei einer Zustandsänderung der Einkaufsliste den Gesamtpreis der neuen Einkaufsliste auf der Konsole aus. Dieser Beobachter besitzt also eine Operation zur Berechnung des Gesamtpreises einer Einkaufsliste.

d) Erstellen Sie eine Klasse mit einer main-Methode, in der Sie eine Einkaufsliste und die zwei entsprechenden Beobachter-Objekte erzeugen. Testen Sie Ihre Implementierung bzgl. der folgenden Operationen auf der erzeugten Einkaufsliste:
   - Add CD3 zum Preis von 25 €
   - Add CD4 zum Preis von 8 €
   - Add DVD2 zum Preis von 145 €
   - Entfernen von CD4
   - Entfernen von CD3
   - Add Buch4 zum Preis von 16 €
   - Add CD4
   - Add Buch 5 zum Preis von 76 €