Übungen zur Vorlesung

Softwaretechnologie

-Wintersemester 2009/2010-

Dr. Günter Kniesel, Pascal Bihler

Übungsblatt 8 - Lösungshilfe

Aufgabe 1. *Proxy-Pattern* (4 Punkte)

a) Erläutern Sie kurz die Motivation und die Funktionsweise des Proxy-Entwurfmusters.

- Absicht
  a. Stellvertreter für ein anderes Objekt
  b. bietet Kontrolle über Objekt-Erzeugung und -Zugriff
- Motivation
  a. kostspielige Objekt-Erzeugung verzögern (zB: große Bilder)
  b. verzögerte Objekterzeugung soll Programmstruktur nicht global verändern

b) Welches sind die teilnehmenden Rollen (Klassen) des Proxy-Entwurfsmusters? Welche Aufgaben übernehmen diese? Welche Operationen sind notwendig?

- Proxy
  a. bietet gleiches Interface wie "RealSubject"
  b. referenziert eine "RealSubject"-Instanz
  c. kontrolliert alle Aktionen auf dem "RealSubject"
- Subject
  a. definiert das gemeinsame Interface
- RealSubject
  a. das Objekt das der Proxy vertritt
  b. eigentliche Funktionalität
- Zusammenspiel
  a. selektives Forwarding

c) Nennen Sie mindestens drei Beispiele für die Anwendung eines Proxy-Entwurfmusters und erläutern Sie kurz die Motivation für den jeweiligen Pattern-Einsatz.

- Virtueller Proxy
  a. verzögerte Erzeugung "teurer" Objekte bei Bedarf
  b. Beispiel: Bilder in Dokument, persistente Objekte
- Remote Proxy
  a. Zugriff auf entferntes Objekt
  b. Objekt-Migration
  c. Beispiele: CORBA, RMI, mobile Agenten
- Concurrency Proxy
  a. nur eine direkte Referenz auf RealSubject

   b. locking des RealSubjects vor Zugriff (threads)
- Copy-on-Write
   a. kopieren erhöht nur internen "CopyCounter"
   b. wirkliche Kopie bei Schreibzugriff und "CopyCounter">0
   c. Verzögerung teurer Kopier-Operationen
- Protection Proxy (Zugriffskontrolle)
   a. Schmaleres Interface
      i. "Kritische" Operationen ausgeblendet
      ii. andere via Forwarding implementiert
   b. ganz anderes Interface
      i. Autorisierung prüfen
      ii. direkten Zugriff gewähren oder Forwarding
   c. Beispiel: "CHOICES" Betriebssystem, JDK

d) Worin unterscheidet sich das „Proxy“-Entwurfsmuster von dem Entwurfsmuster „Adapter“?

- Proxy
   a. gleiches Interface
   b. kontrolliert Zugriff
- Adapter
   a. verschiedenes Interface

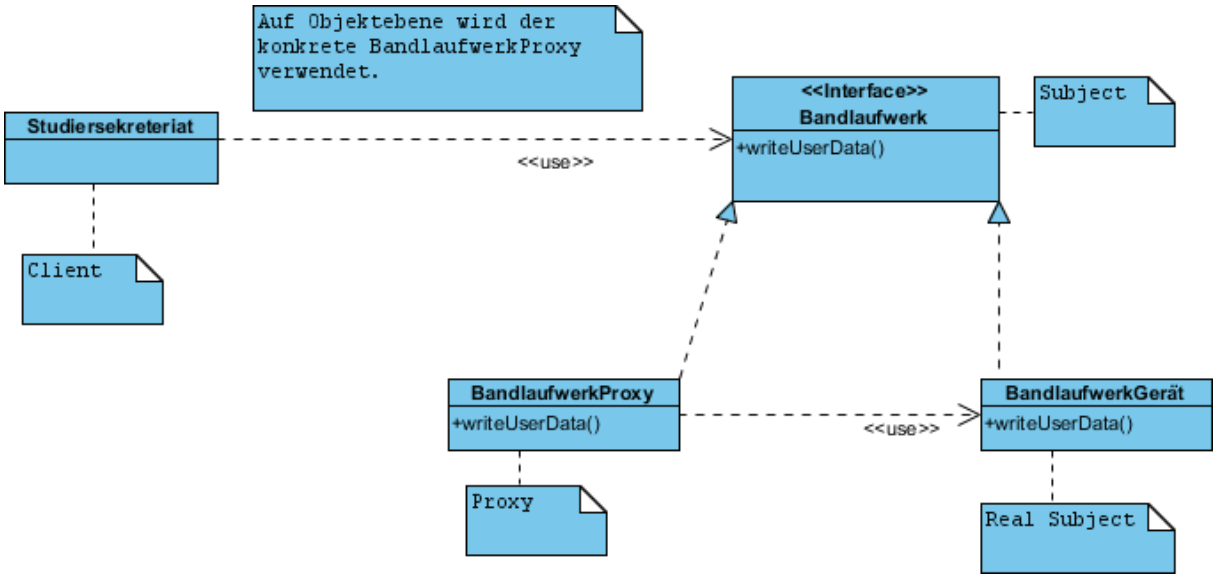
Aufgabe 2. *Entwurfmuster im Einsatz* (10 Punkte)

a) Im Studiensekretariat wird ein Bandlaufwerk für die langfristige Speicherung von Daten genutzt. Jeden Tag wird ein Backup dieser für das Prüfungsbüro wichtigen Daten gemacht. Ein Schreibvorgang der Daten dauert hierbei einen erheblichen Zeitraum (> 60 Minuten), da das Gerät keinen hohen Schreibdurchsatz im Vergleich zu modernen Systemen erreicht. Das Studiensekretariat ist aber auf dieses Gerät angewiesen und stellt Ihnen die Aufgabe, eine Möglichkeit zu finden, das Gerät ohne Änderung der verwendeten Schnittstelle weiter zu verwenden und

   a. Schreibvorgänge nur zwischen 22:25 und 05:45 Uhr zu erlauben.

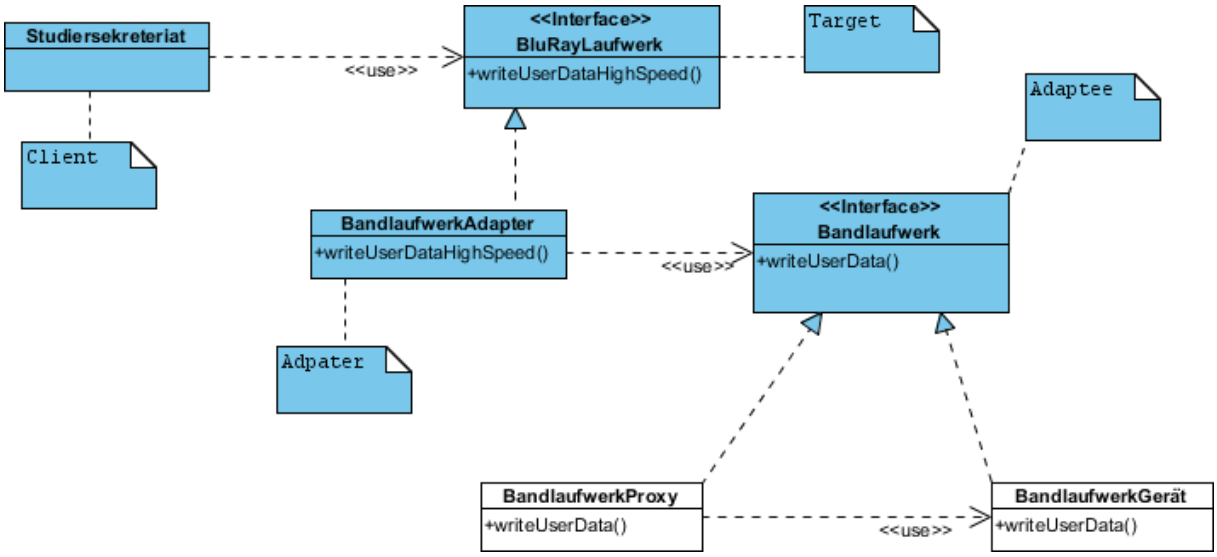
   b. Schreibvorgänge außerhalb dieses Zeitfensters oder während eines laufenden Schreibvorgangs in einer Warteschlange zwischen zu speichern und verzögert auszuführen.

   Zeichnen Sie für Ihre Lösung des Problems ein Klassendiagramm. Erläutern Sie wichtige Rollen (Klassen, Methoden) in Form von Notizen.



b) Eine neue Version der Sekretariats-Software unterstützt keine Bandlaufwerke mehr, sondern nur noch die Hochgeschwindigkeits-Sicherung auf Blu-ray Discs. Wie müssen Sie die in a) entworfene Architektur anpassen, um hohe Investitionskosten zu vermeiden und das Bandlaufwerk weiter verwenden zu können? Welches Entwurfsmuster setzen Sie ein? Erläutern Sie wieder die Rollen in Form von Notizen.

Adapter-Pattern:

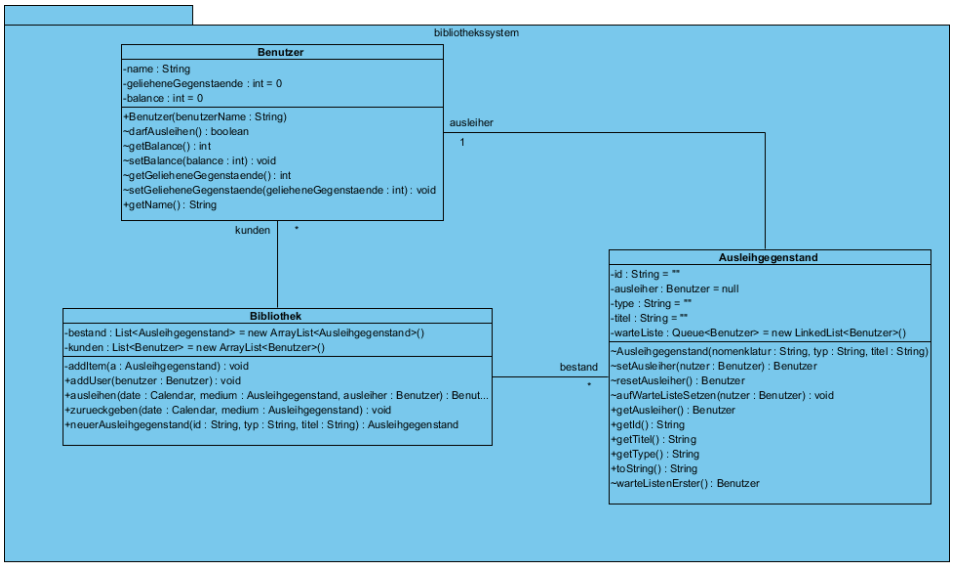


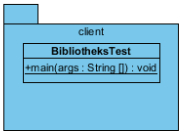
Aufgabe 3. *Modell-Generierung* (3 Punkte)

Gegeben sei der Kern einer Bibliotheksverwaltung, deren Java-Programmode Sie aus Ihrem Tutoriums-SVN-Repository auschecken können.

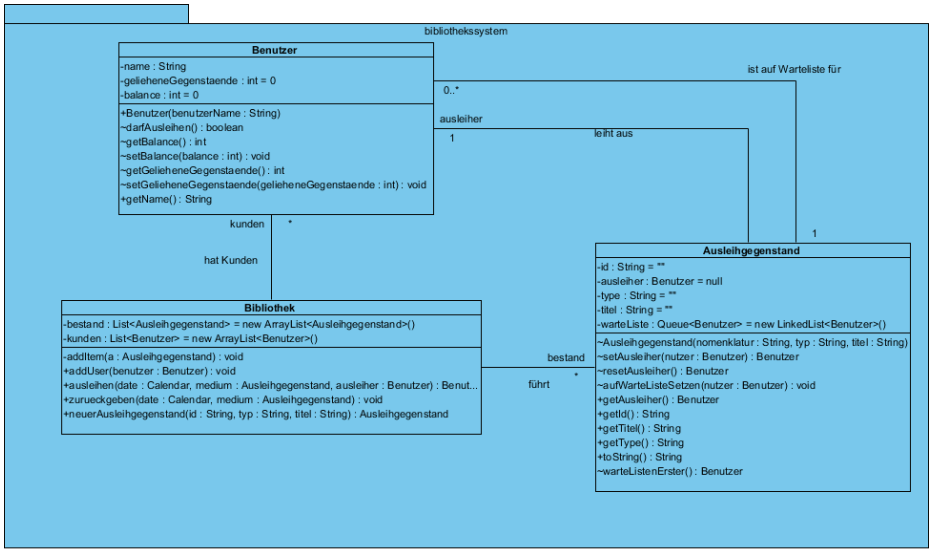
a) Wählen Sie aus dem Kontextmenü des Projektes den Menüpunkt „Open SDE-EC“ aus und wechseln Sie zur Modell-Perspektive. Wählen Sie aus dem Menü „Modeling“ den Befehl „Instant Reverse…“ und fügen Sie als „Source Folder“ den „src“-Ordner des Projektes hinzu. Nach Klick auf „OK“ wählen Sie in der „Class Repository“ View die importierten

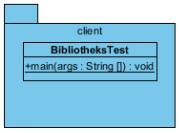
Pakete aus und wählen aus dem Kontextmenü „Reverse Resources to → New Class Diagram“. Speichern Sie das Ergebnis in Ihr SVN-Repository.





b) Erörtern Sie, ob die ebenfalls generierten Assoziationen einer sinnvollen Modellierung entsprechen und geben Sie den Assoziationen aussagekräftige Namen. Ergänzen Sie fehlende Assoziationen.





Aufgabe 4. *Entwurfsmuster* (13 Punkte)

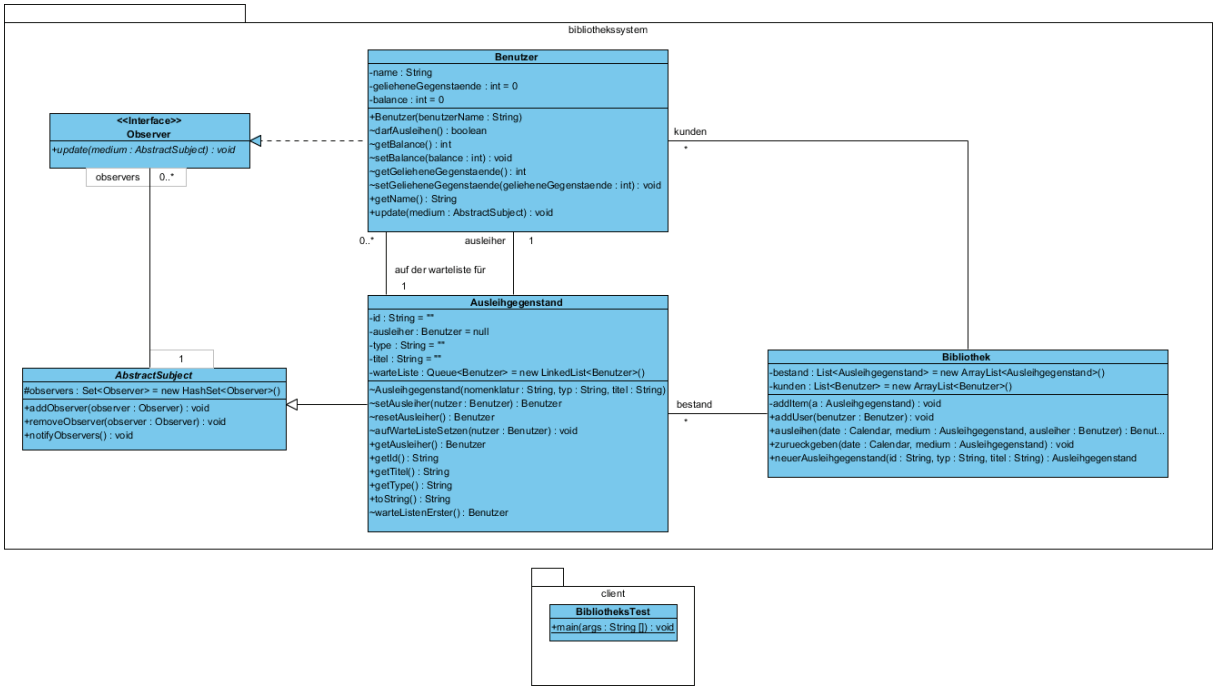
Ziel ist es nun, die Architektur des obigen Minimal-Systems an geeigneter Stelle mittels Design Patterns zu erweitern.

Hinweis: Der mitgelieferte Programmcode enthält im Paket `client` die Klasse `BibliotheksTest`. Benutzen Sie diese als Hauptprogramm, um Ihre Implementierung zu testen.

a) In unserem System führt jedes Objekt der Klasse Ausleihgegenstand eine Warteliste, auf der Kunden sich bei Interesse für ein aktuell entliehenes Medium eintragen können. Realisieren Sie mittels eines geeigneten Patterns, dass der erste Kunde aus der Warteliste eines Mediums informiert wird, sobald dieses Medium zurückgegeben wurde. Als Reaktion auf die Benachrichtigung soll das Kunden-Objekt eine E-Mail an den entsprechenden Kunden schicken (symbolisiert durch die Text-Ausgabe „Schicke E-Mail an *...Name...*: *... Inhalt...*“).

```
public interface Observer {

	void update(AbstractSubject medium);

}


Public abstract class AbstractSubject {
	protected Set<Observer> observers = new HashSet<Observer>();

	public void addObserver(Observer observer) {
		if (observer == null)
			return;
		observers.add(observer);
	}

	public void removeObserver(Observer observer) {
		if (observer == null)
			return;
		observers.remove(observer);
	}

	public void notifyObservers() {
		for(Observer observer:observers) {
			observer.update(this);
		}
	}
}

public class Benutzer implements Observer{

…

	@Override
	public void update(AbstractSubject medium) {
		System.out.println("Schicke E-Mail an " + getName() +
		": Das Medium '" + ((Ausleihgegenstand) medium).getTitel() +
		"' ist verfügbar!");
	}
}

public class Ausleihgegenstand extends AbstractSubject {
```

```
        …
        Benutzer setAusleiher(Benutzer nutzer){
                …

                if(warteListe.peek()==nutzer){
                        removeObserver(nutzer);
                        warteListe.poll();
                        addObserver(warteListenErster());

                        …
                } …
        }

        Benutzer resetAusleiher(){
                ausleiher = null;
                notifyObservers();
                …
        }

        void aufWarteListeSetzen(Benutzer nutzer) {
                if(!warteListe.contains(nutzer)){
                        warteListe.offer(nutzer);
                        addObserver(warteListenErster());
                }
        }

        …
}
```



b) In einem nächsten Schritt soll das E-Mail-System der Bibliothek erweitert werden. Erstellen Sie dazu eine Klasse `EmailSystem`, die das E-Mail-System repräsentiert und eine Methode `sendeMail(String name, String nachricht)` unterstützt (die eine Zeichenkette auf `System.out` ausgibt). Garantieren Sie durch die korrekte Anwendung eines geeigneten Entwurfsmusters, das maximal eine Instanz Ihrer Klasse erzeugt und verwendet wird.
Binden Sie zuletzt die Klasse und ihre Funktionalität in das Gesamtsystem ein.

```
package bibliothekssystem;

public class EmailSystem {

	protected static EmailSystem instancePool;

	private EmailSystem() {
	// private! Verhindert die direkte Generierung der Klasse
		System.out.println("Initialisiere E-Mail-System");
	}

	public void sendeMail(String name, String nachricht) {
		System.out.println("Schicke E-Mail an " + name + ": "
					+ nachricht);
	}

	protected static EmailSystem getInstance() {
		if (instancePool == null)
			instancePool = new EmailSystem();
		return instancePool;
	}

}

public class Benutzer implements Observer{

…

	@Override
	public void update(AbstractSubject medium) {
		EmailSystem.getInstance().sendeMail(getName(), "Das Medium '" +
		((Ausleihgegenstand) medium).getTitel() + "' ist verfügbar!");
	}
}
```