Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Institut für Programmstrukturen und Datenorganisation (IPD)

Prof. Dr. Walter F. Tichy

Vorlesung Softwaretechnik 1

Übungsblatt 2
Sommersemester 2009

Ausgabe: 11.05.2009

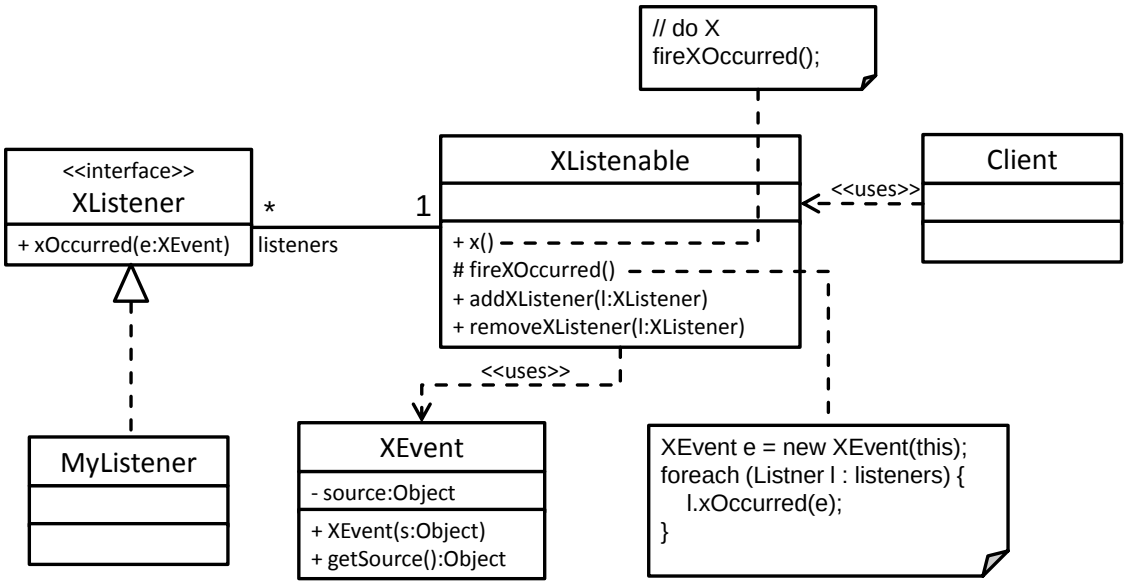
Abgabe: 20.05.2009, 11:00 Uhr

Achtung: Abgabe bis Mittwoch, den 20.05.2009, 11:00 Uhr!

UML-Diagramme, Swing

Aufgabe 1: UML-Sequenzdiagramme (4 Punkte)

Gegeben sei folgendes UML-Klassendiagramm:

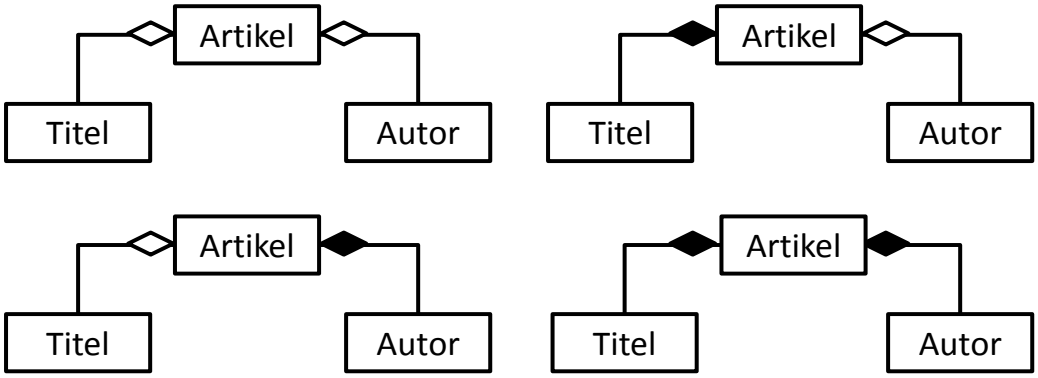


Stellen Sie folgenden Sachverhalt in einem Sequenzdiagramm dar:

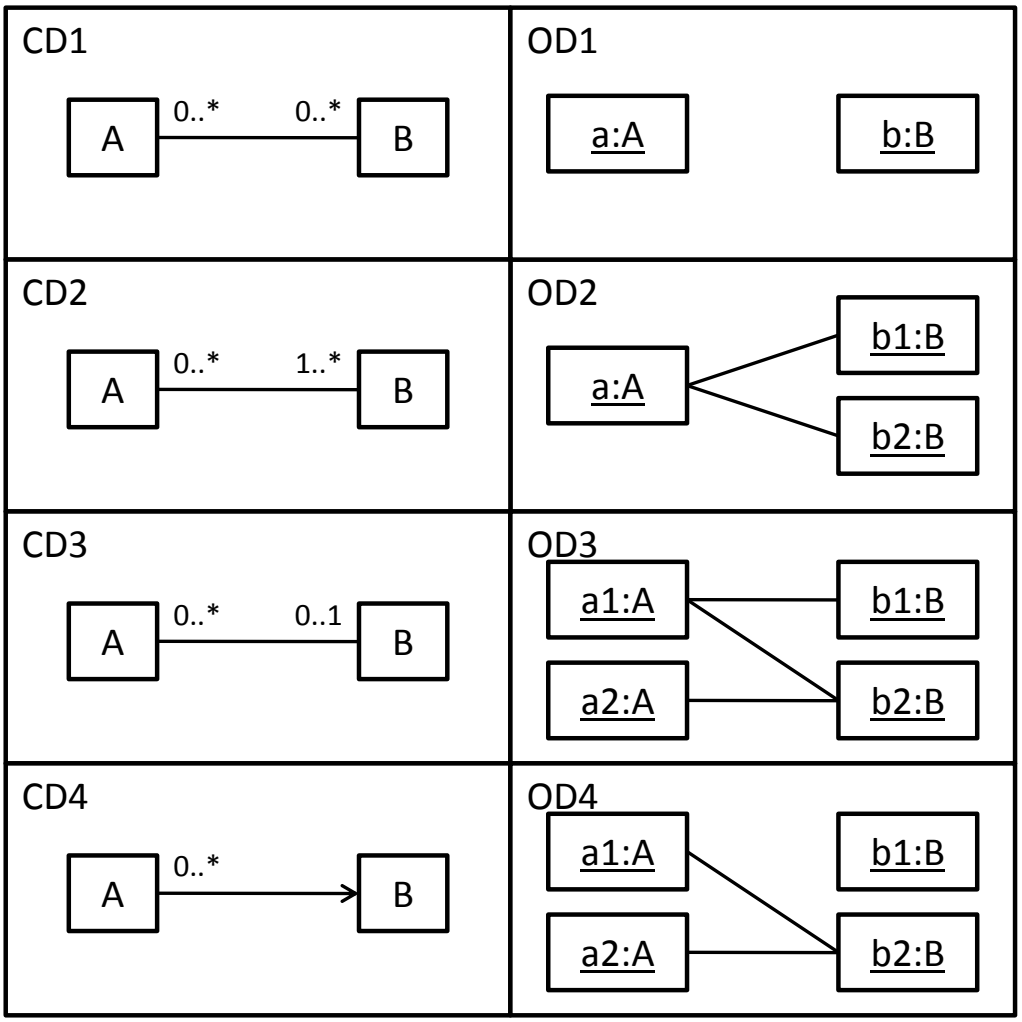
> *Ein Objekt l der Klasse MyListener möchte exakt ein Mal darüber informiert werden, wenn in dem Objekt x vom Typ XListenable die Methode x() aufgerufen wird. Dazu meldet sich das Objekt l als Beobachter bei Objekt x an. Nach unbestimmter Zeit ruft ein Objekt der Klasse Client bei Objekt x die Methode x() auf. Daraufhin wird das Objekt l durch das Senden eines XEvents e über das Auftreten von x() informiert, greift auf die Quelle des Ereignis-Objekts e zu und meldet sich direkt im Anschluss als Beobachter von x ab.*

Aufgabe 2: Semantik von UML (6 Punkte)

a) Welche der folgenden Modellierungen ist richtig, wenn man geeignete Multiplizitäten ergänzt? Markieren Sie die richtige Modellierung und ergänzen Sie sinnvolle Multiplizitäten. Begründen Sie ihre Entscheidung. Ohne Begründung gibt es keine Punkte. (2 Punkte)



b) Geben Sie für jede Kombination der Objekt- und Klassendiagramme an, ob sie zusammen passen oder nicht. Wählen Sie zur Darstellung der Zugehörigkeit von Objektdiagramm zu Klassendiagramm eine geeignete Darstellung. (4 Punkte)



Aufgabe 3: Parametervarianz in Java (6P)

Java kennt nur Rückgabewerte und Eingabeparameter. Welche Varianz wäre nach dem Substitutionsprinzip für diese beiden Parameterarten zulässig? Welche Varianz ist bei den beiden Parameterarten tatsächlich bei vererbten Methoden zulässig, welche nicht? Untersuchen Sie jeweils Ko- und Kontravarianz. Geben Sie (möglichst wenig) Java-Quelltext an, der geeignet ist, Ihre Aussagen zu be-

legen. Erläutern Sie wie/warum das angegebene Code-Stück Ihre Aussagen belegt – ggf. auch durch Angabe von Übersetzerfehlermeldungen.

Hinweis: Wir stellen häufig eine „Oberfrage“ und nachfolgend viele kleine „Unterfragen“. Diese untergeordneten Fragen sollen Ihnen helfen, Ihre Antwort zu strukturieren und Sie auf die richtige Fährte zu locken, damit Sie die übergeordnete Frage im Sinne der Aufgabe beantworten. Beispielsweise in dieser Aufgabe hier gibt es aber mehrere „Oberfragen“. Die müssen alle beantwortet werden, um volle Punktzahl zu erreichen. **Wenn *mehrere Fragen* in einer Aufgabe stehen, beantworten Sie daher stets *alle* Fragen.** Tipp: Haken Sie die Fragen (und Anweisungen) einzeln ab. Das klappt dann auch gut in der Klausur!

Aufgabe 4: Swing - Internationalisierung (10 Punkte)

Erstellen Sie eine Java-Anwendung mit einer grafischen Swing-Benutzeroberfläche. Das Fenster der Java-Anwendung soll aus einer Menü- (`JMenuBar`) und Werkzeugleiste (`JToolBar`) bestehen. Die Menüleiste soll mindestens 2 Menüs (`JMenu`) enthalten, welche wiederum jeweils mindestens 3 Menüeinträge (`JMenuItem`) enthalten. Die Menüs müssen einen Namen und einen mnemonischen Buchstaben haben. Die Menüeinträge müssen ein Namen, ein Icon, ein Tastaturkürzel, ein mnemonischen Buchstaben sowie einen Tooltip besitzen. Die Werkzeugleiste muss genau die gleichen Funktionen wie die Menüleiste anbieten. Die Einträge in der Werkzeugleiste müssen ein Icon und ein Tooltip besitzen. Es muss erkennbar sein, welcher Menüeintrag zu welchem Werkzeugleisteneintrag gehört.

Wird ein Menüeintrag bzw. ein Knopf in der Werkzeugleiste gedrückt, soll auf der Konsole eine kurze Nachricht (≠ Name des Menüeintrags/Knopfes) ausgeben werden, die beschreibt welche Funktion ausgelöst wurde. Die Namen und Nachrichten dürfen nicht trivial sein (also nicht „Knopf 1“, „Knopf 2“, …).

Die obige Java-Anwendung soll anschließend internationalisiert werden. Sie müssen mindestens die Sprachen Deutsch und Englisch unterstützen. Es müssen dabei alle Texte der Java-Anwendung internationalisiert werden. Icons und Tastaturkürzel müssen Sie nicht internationalisieren. Die gewünschte Sprache soll bei Programmstart als Parameter übergeben werden können. Verwenden Sie in Ihrem Java-Quelltext einen einheitlichen Programmierstil, wie Sie es in der Vorlesung „Programmieren I“ bei Prof. Snelting gelernt haben. Wir werden den Programmierstil diesmal stärker in die Bewertung einbeziehen.

Hinweis: **Bitte lesen Sie sich den folgenden Text aufmerksam durch. Missachtung der Anweisungen wird Sie Punkte kosten!**

Den Quelltext Ihres Programmes drucken Sie bitte aus und geben ihn zusammen mit dem Übungsblatt ab. Zusätzlich müssen Sie sowohl den Quelltext als auch die Java-Anwendung an folgende E-Post-Adresse senden:

swt1@ipd.uka.de

Als Betreff geben Sie bitte **[Uebungsblatt 2] <TUTNR> <MATRNR> <VORNAME> <NACHNAME>** an.

Hängen Sie den Java-Quelltext als ZIP-Archiv an die E-Mail an. Der Name des ZIP-Archivs muss folgendes Format haben: **<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.zip**. Das ZIP-Archiv darf nur den Java-Quelltext (d.h. Dateien mit der Endung .java) und evtl. benötigte Ressourcen (wie bspw. Icons) enthalten. Sofern Sie mit Paketen gearbeitet haben, muss diese Ordnerstruktur ebenfalls in Ihrem ZIP-Archiv vorhanden sein.

Verwenden Sie in Ihrem Java-Quelltext nur ASCII-Zeichen und vermeiden Sie IDE-spezifische Java-Annotationen. Die lauffähige Java-Anwendung hängen Sie als **ausführbares JAR-Archiv** zusätzlich zum ZIP-Archiv an die E-Mail an. Es werden nur Java-Anwendungen akzeptiert, welche als ausführbares JAR-Archiv an die E-Mail angehängt sind! Das JAR-Archiv muss als separater Anhang gesendet werden. Der Name des JAR-Archivs muss wie folgt lauten: **<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.jar**

Beispiel: Sie heißen Hans Müller, gehen in das Tutorium 7 und Ihre Matrikelnummer ist 123456. Ihr Betreff sieht dann wie folgt aus:

[Uebungsblatt 2] 07 123456 Hans Mueller

Die Anhänge heißen dann:

07_123456_Hans_Mueller.zip / 07_123456_Hans_Mueller.jar

Die E-Mail mit beiden Anhängen muss bis zum offiziellen Abgabetermin bei uns eingegangen sein. Vergewissern Sie sich, dass Sie alle Code-Dateien, die zur Übersetzung der Java-Anwendung notwendig sind, mitschicken und der Quelltext übersetzt werden kann. Prüfen Sie, dass die Java-Anwendung aus dem JAR-Archiv lauffähig ist. Andere Dateiformate werden nicht akzeptiert.

Bei Missachtung der obigen Anforderungen oder Teilen davon wird die gesamte Aufgabe mit 0 Punkte bewertet.

Tipp: Geeignete frei verfügbare Icons gibt es beispielweise von Sun http://java.sun.com/developer/techDocs/hi/repository/ oder dem Tango Desktop Project http://tango.freedesktop.org/Tango_Icon_Library.