

1 Grundlagen der Objektorientierten Programmentwicklung

Punkte

- 1.1 Bei Referaten und Seminararbeiten müssen die verwendeten Quellen angegeben werden. Üblicherweise werden Quellen nach folgendem Muster zitiert:

Name des Autors/der Autorin Erscheinungsjahr: Titel.
Erscheinungsort: Verlag.

Bsp.: Weißalles, Willi 2004: IT-Abitur fast geschenkt.
Mannheim: Schlaumeier.

Entsprechend dieser Anforderungen wurde die Klasse *Quellenangabe* modelliert. Damit lassen sich jedoch keine Quellen aus Zeitungsartikeln angeben. Denn dabei werden folgende Angaben gefordert:

Name des Autors/der Autorin Erscheinungsjahr: Titel des Artikels.
Name der Zeitung Jahrgang (Ausgabe): Seitenzahl.

Bsp.: Weißalles, Willi 2004: Was fange ich mit dem Abitur an?..
IT-News 133 (24. Oktober): 17.

Quellenangabe
derTitel : Text
derAutor : Text
dasJahr : GZ
derVerlag : Text
derOrt : Text
...
+ druckeQuelle()

Erweitern Sie die objektorientierte Software um diese Fähigkeit. Verwenden Sie dazu die Klasse *Quellenangabe*. Die Klasse *Quellenangabe* darf nicht verändert werden. Stellen Sie Ihre Lösung in UML-Notation dar.

2 Leihbücherei

Eine einfache Software für eine Leihbücherei soll programmiert werden. Büchern und Lesern werden eindeutige fortlaufende Nummern zugewiesen. Beim Ausleihen oder der Rückgabe eines Buches reicht es daher aus, die Buch- und die Lesernummer einzugeben. Wird ein Buch aus-sortiert, darf die Nummer nicht erneut vergeben werden. Wenn ein Buch verliehen wird, wird ein Ausleiheintrag erstellt, in dem die Buchnummer, die Leser-nummer und das Ausleihdatum festgehalten werden. Diese Daten können in eine Datenbank über-tragen werden, um daraus Ausleih-Statistiken zu erstellen.

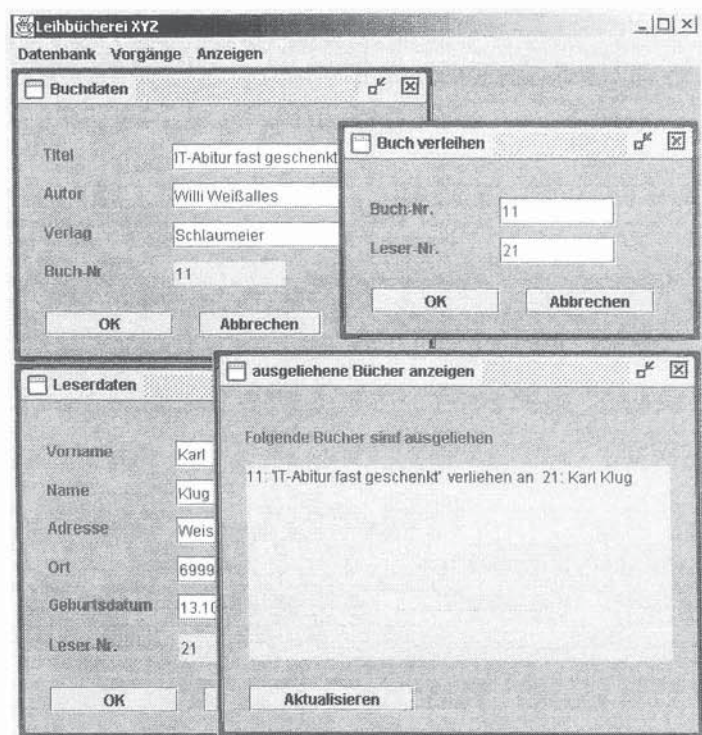
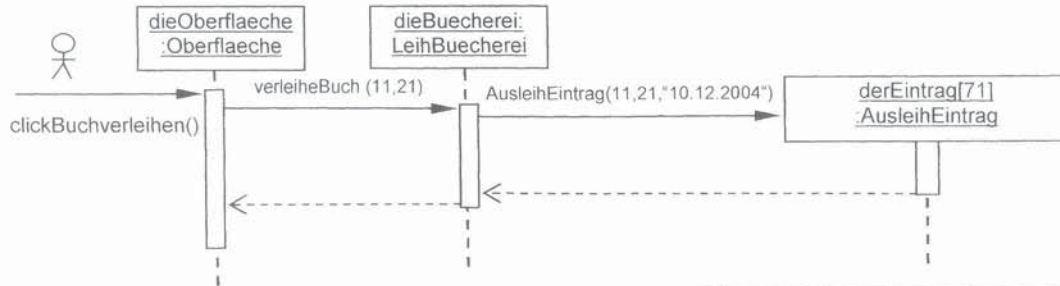


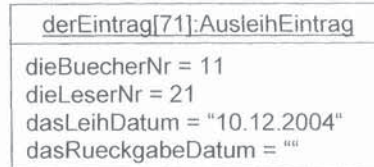
Abbildung 1: Oberfläche der Leihbücherei-Software

Zu Abbildung 1: Der Anwender hat mehrere Fenster geöffnet. Die internen Fenster werden durch das Objekt der Klasse *Oberflaeche* verwaltet. Die Steuerung der Fenster ist nicht Gegenstand dieser Aufgabe.

- 2.1 Das folgende Sequenzdiagramm verdeutlicht die Vorgänge beim Verleihen eines Buches. Szenario: "Der Leser mit der Nummer 11 leiht am 10.12.2004 das Buch mit der Nummer 21 aus". Der Anwender hat die Buch-Nr. und die Leser-Nr. in das Fenster eingetragen und klickt auf den OK-Knopf (siehe Abbildung 1).

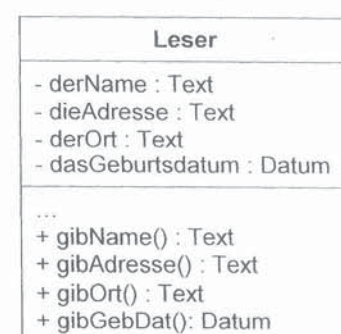
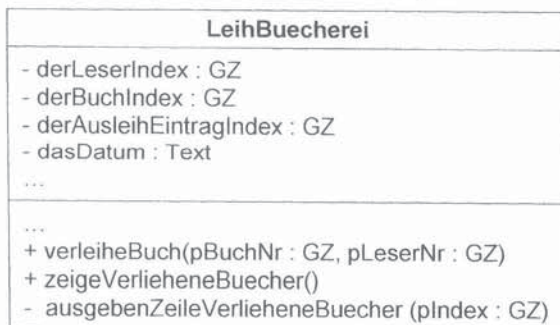
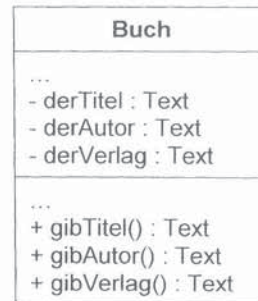
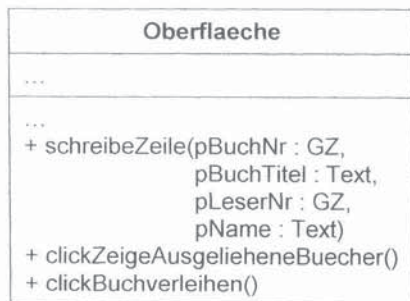


Es handelt sich dabei um den 72. Ausleihvorgang. Durch diesen Vorgang entsteht folgendes Objekt:



Erstellen Sie die vollständige Klasse zum Objekt *derEintrag[71]* in UML-Notation. Geben Sie alle benötigten Attribute und Operationen an. Tragen Sie Sichtbarkeiten, Parameter, Rückgabewerte und den Konstruktor ein.

- 2.2 Folgende weitere Klassen sind für die Modellierung der Software ausreichend:



Konstruktoren und Destruktoren sind nicht dargestellt.

Die Buch- und Leser-Objekte werden in Feldern verwaltet. Die Buch- bzw. Leser-Nr. (siehe Abbildung 1) entspricht dabei jeweils dem Feldindex.

Die Operation *ausgebenZeileVerlieheneBuecher()* der Klasse *LeihBuecherei* sammelt alle erforderliche Daten, die in einer Zeile des Fensters "ausgeliehene Bücher anzeigen" (siehe Abbildung 1) angezeigt werden sollen.

Die Operation *schreibeZeile()* der Klasse *Oberflaeche* schreibt den übergebenen Text in das Fenster "ausgeliehene Bücher anzeigen".

Punkte

Stellen Sie die Assoziationen zwischen den Klassen in einem Klassendiagramm dar. Attribute und Operationen brauchen nicht dargestellt werden. Verwenden Sie die Klassen, die oben dargestellt sind und die Klasse *AusleihEintrag* aus Aufgabe 2.1. Geben Sie bei den Assoziationen die Kardinalitäten und Rollennamen an. Begründen Sie jeweils kurz, weshalb Sie sich für die eingetragene Kardinalität und Richtung der Assoziation entschieden haben.

- 2.3 Modellieren Sie das folgende Szenario in Form eines Sequenzdiagramms:
"Der Anwender wählt aus dem Menü 'Anzeigen' den Menübefehl 'Ausgeliehene Bücher anzeigen'"
Die Daten aller Ausleiheinträge werden daraufhin in das Ausgabefeld des Fensters 'ausgeliehene Bücher anzeigen' geschrieben (siehe Abb. 1). Von jeder Ausleiheung wird die Buchnummer, der Buchtitel, die Lesernummer und der Lesername angegeben.

7

3 Zustandsdiagramm eines Buch-Objektes

4

Ein Zustandsdiagramm für ein Buch-Objekt soll erstellt werden. Die Modellierung ist von der Aufgabe 2 unabhängig.

Wenn in der Bücherei ein Buch beschafft wird, werden seine Daten erfasst, und ein neues Objekt der Klasse *Buch* wird erzeugt. Die Bücher können von den Lesern ausgeliehen werden. Wird ein ausgeliehenes Buch von einem anderen Leser gewünscht, dann kann er es vorbestellen. Bücher, die nicht vorbestellt wurden, stehen nach der Rückgabe sofort für eine neue Ausleihe bereit. Vorbestellte Bücher werden nach der Rückgabe für den entsprechenden Leser zur Abholung bereitgelegt. Wird das Buch nicht fristgemäß vom benachrichtigten Leser abgeholt, dann steht es für eine neue Ausleihe zur Verfügung.

Verlorene oder defekte Bücher werden bei der Modellierung nicht berücksichtigt.



Ein neues im System eingegebenes Buch befindet sich zunächst im Zustand *ausleihbar*.

Erstellen Sie ein Zustandsdiagramm für ein Buch-Objekt. Transitionen können durch freie Texte beschrieben werden.

4 Algorithmus zur Berechnung der Ausleihdauer

7

Entwickeln Sie den Algorithmus der Operation

berechneTage (pDat1: Datum, pDat2: Datum) : GZ.

Diese Operation dient dazu, die Differenz an Tagen zwischen zwei eingegebenen Daten zurückzugeben. Zur Vereinfachung sei vereinbart:

- der 1. Parameter – eine Instanz der Klasse *Datum* - übergibt immer das ältere Datum
- berücksichtigen Sie nur einen Jahreswechsel
- berücksichtigen Sie keine Schaltjahre.

Datum
- derTag : GZ
- derMonat : GZ
- dasJahr : GZ
...
+ gibTag() : GZ
+ gibMonat() : GZ
+ gibJahr() : GZ

Verwenden Sie die Operationen der Klasse *Datum*, welche die Tages-, Monats- und Jahreswerte als Ganzzahlen zurückgeben.

Beispiel: *datum1* besitzt als Wert das Datum des 19. April 2004
datum2 besitzt als Wert das Datum des 30. Juni 2004
berechneTage (datum1, datum2) berechnet die Differenz von 72 Tagen.

Beispiel: *datum1* besitzt als Wert das Datum des 15. Dezember 2004
datum2 besitzt als Wert das Datum des 21. Februar 2005
berechneTage (datum1, datum2) berechnet die Differenz von 68 Tagen.

Verwenden Sie das Feld *dieTageImMonat[13]*: GZ, in dem die Anzahl der Tage der einzelnen Monate abgelegt ist. Das Feld ist mit folgenden Werten belegt:
 {0,31,28,31,30,31,30,31,31,30,31,30,31}

Beispiel: die Anzahl der Tage des Monats Januar (31 Tage) lassen sich ermitteln mit
anz <- dieTageImMonat[1].

Punkte

30