

Schriftliche Abiturprüfung 2014

Fach:	Informatiksysteme (Fachrichtung Technik)
Kurstyp:	E-Kurs
Datum:	05. Mai 2014
Bearbeitungszeit:	5 Zeitstunden
Hilfsmittel:	Java-IDE Netbeans, Umllet (Tool zum Zeichnen des Klassendiagramms), XAMPP (MySQL-DBMS, SQL-Client phpmyadmin)
Seitenzahl:	Die Prüfungsaufgabe umfasst mit Deckblatt 9 Seiten.

Einleitung:

Einem Teil der hier zu bearbeitenden Aufgaben liegt die im Folgenden beschriebene Situation zugrunde.

Ein Hotel will zur Verwaltung von Gästen und Zimmerbuchungen eine Softwareanwendung einsetzen. Als Stammdaten werden die verfügbaren Zimmer unterschiedlicher Kategorie (Einzelzimmer, Doppelzimmer, Familienzimmer/Suite) eingepflegt und die Gäste bzw. Kunden aufgenommen und gespeichert.

In einer ersten Version sollen Buchungen angelegt werden können, wobei den Gästen Zimmer zugeordnet werden. Die Anzahl und Art der verfügbaren freien Zimmer und die aktuell anwesenden Gäste sollen jederzeit angezeigt werden können.

Die Objektorientierte Analyse des Auftrags ergab folgende Grundlage für das Klassendiagramm als Grundlage für die Implementierung.

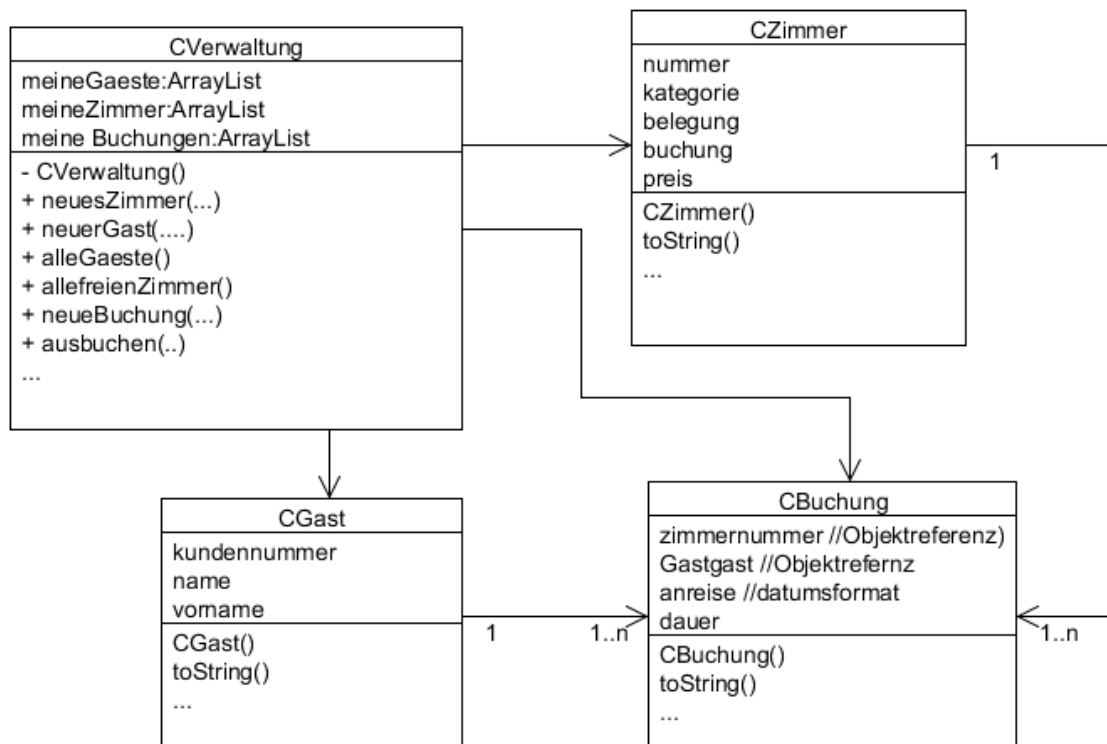


Abbildung 1

Bearbeiten Sie in dem oben beschriebenen Kontext einige Aufgaben zu zentralen Themen des Fachs Informatiksysteme, für die Sie die Rolle eines Mitarbeiters einer Firma für Softwareentwicklung übernehmen sollen. Ihre Entwicklungswerkzeuge sind die Programmiersprache Java und die Modellierungssprache UML.

Die genauen Anforderungen des Projekts entnehmen Sie bitte dem Auszug des **Pflichtenheftes** und den detaillierten Aufgabenstellungen unten.

Zielbestimmung

Im ersten Schritt soll die Stammdatenpflege (Zimmer, Kunden) realisiert werden. Hierfür sollen zunächst alle notwendigen Methoden entwickelt werden. Die Entwicklung der zur Stammdatenpflege vorgesehenen Dialoge erfolgt zu einem späteren Zeitpunkt. Ebenso wird die Datenhaltung mit einem entsprechenden Datenbanksystem erst zu einem späteren Zeitpunkt realisiert.

Produktfunktionen

- /F10/ Abspeichern der Stammdaten in Containern.
- /F20/ Abändern der Stammdaten über entsprechende Methoden.
- /F30/ Löschen von Stammsätzen aus den vorgesehenen Listen.
- /F40/ Auflisten aller Stammdaten(Kunden, Zimmer).
- /F50/ Buchungsvorgänge durchführen und speichern.
- /F60/ Auflisten aller Buchungen.
- /F70/ Eine Konsolenanwendung zum Testen aller implementierten Methoden.

Produktdaten

- /D10/ Es können beliebig viele Stammdatensätze angelegt werden.
- /D20/ Der Stammdatensatz eines Kunden/Gastes umfasst alle oben im Klassendiagramm angegebenen Attribute.
- /D30/ Der Stammdatensatz eines Zimmers umfasst alle oben im Klassendiagramm angegebenen Attribute.
- /D40/ Eine Buchung bezieht sich immer auf ein Zimmer und speichert Anreise und Dauer des Aufenthalts.
- /D50/ Ein Gast/Kunde bucht immer ein Zimmer. Es können natürlich mehrere Buchungen für unterschiedliche Zimmer zur gleichen Zeit getätigt werden.

Benutzungsoberfläche

- /B10/ Zu Testzwecken zunächst zeichenorientiert.

Auf Grundlage des Plichtenheftes wird für das Software-System schließlich das sogenannte Fachkonzept erstellt. Dort werden die im Plichtenheft formulierten Produktfunktionen nochmals spezifiziert.

Aufgabe 1

Allgemeine Fragen zur Objektorientierten Anwendungsentwicklung

Zentrale Begriffe in der Objektorientierten Programmierung sind **Klasse** und **Objekte**. Objekte werden durch ihre Attribute charakterisiert und tauschen untereinander Botschaften aus. In diesem Kontext fällt häufig der Begriff Multiplizität.

a Multiplizität

- i Was versteht man unter der Multiplizität einer Assoziation zwischen Klassen in der Objektorientierten Programmierung?
- ii Welche Art der Assoziation bezogen auf die Multiplizität liegt zwischen den Klassen CGast und CBuchung im Beispiel der Abbildung 1 vor?
- iii Nennen und erläutern Sie kurz die beiden anderen möglichen Multiplizitäten, die es zwischen Klassen geben kann? Konstruieren Sie jeweils ein Beispiel mit entsprechender UML-Notation!
- iv Zwischen den Klassen CGast und CZimmer bildet CBuchung eine Assoziationsklasse. Was versteht man darunter?

b Schnittstellen (Interfaces)

- i Was versteht man in der Objektorientierten Programmierung unter einem Interface? Was ist seine Bedeutung und in welchem Kontext findet es in der OOP häufig Verwendung?
- ii Viele Container implementieren das häufig verwendete Interface Iterator, so zum Beispiel die Klasse ArrayList. Schreiben Sie eine Anwendung, die eine Liste mit Strings (Namen) durchläuft und ausgibt. Verwenden Sie die Methoden hasNext() und next().

c Sortieren

- i Beschreiben Sie das Sortierverfahren Insertionsort (Sortieren durch Einfügen) in Pseudocode.
- ii Führen Sie das Verfahren am Beispiel der Zahlenfolge in Abbildung 2 durch.

0	1	2	3	4	5	6	7	8	9
88	99	19	25	98	10	23	40	27	62

Abbildung 2

Aufgabe 2

Objektorientierte Programmentwicklung

- a Vervollständigen Sie das in Abbildung 1 vorgegebene unvollständige Klassendiagramm.
- b Implementieren Sie auf Grund der in Einleitung und Pflichtenheft formulierten Vorgaben sowie Ihrem in Aufgabe 2 a erstellten UML-Klassenmodell die Klassen CGast , CZimmer und CBuchung.
- c Implementieren Sie in allen Klassen die Methode toString() im Hinblick auf eine spätere Ausgabe auf der Konsole.
- d Verwenden Sie in CBuchung einen Datumstyp für das Attribut *ankunft*. Verwenden Sie als Eingabeparameter für das Datum in den Methoden den Typ String und wandeln Sie den Typ entsprechend um.
- e Verwenden Sie in CBuchung ferner Objektreferenzen auf CGast und CZimmer, um eine Navigierbarkeit von CBuchung zu CGast und CZimmer zu gewährleisten.
- f Die Kategorien der Zimmer(A,B,C) sollen als Einzel-, Doppel- und Familienzimmern (Suiten) ausgegeben werden. Berücksichtigen Sie dies in den Methoden.

Aufgabe 3

- a Ergänzen Sie die Anwendung um die Containerverwaltung der Gäste, Zimmer und Buchungen **CVerwaltung** gemäß der Vorgabe im Klassendiagramm.
- b Bei folgender Klasse **CVerwaltung** soll sichergestellt werden, dass nur **genau ein Objekt** erzeugt werden kann (**Singleton Muster**).

Zu den Methoden in CVerwaltung:

- neuerGast(...) erzeugt einen neuen Gast und fügt ihn in die ArrayList ein. Die Kundennummer soll generiert werden.
- neuesZimmer(...) erzeugt ein neues Zimmer mit Nummer, Kategorie und Preis.
- alleGaeste() erzeugt eine Ausgabe aller Gäste, die gespeichert sind.
- AlleaktuellenGaeste() erzeugt eine Ausgabe aller Gäste, die gebucht haben.
- allefreienZimmer() erzeugt eine Ausgabe aller freien Zimmer.
- alleZimmer() erzeugt eine Ausgabe aller vorhandenen Zimmer.
- buchen(...) erzeugt eine Buchung eines Zimmers mit Angabe der Ankunft und der Dauer.
- ausbuchen() gibt das Zimmer frei, und eine Rechnung wird erstellt.

a Programmieren der Testfälle

Erstellen Sie eine Testklasse MyHotelUI, in der folgender Test nachgestellt wird:

- Legen Sie drei Doppelzimmer und zwei Familienzimmer (Suiten) an.
- Geben Sie eine Liste der angelegten Zimmer auf der Konsole aus.
- Legen Sie zwei Gäste an.
- Geben Sie eine Liste aller Gäste auf der Konsole aus.
- Buchen Sie für einen Gast ein Doppel- und ein Familienzimmer.
- Buchen Sie für den zweiten Gast ein Doppelzimmer.
- Geben Sie die freien Zimmer wieder aus.
- Buchen Sie einen der Gäste aus und geben Sie den Rechnungsbetrag an.

Die Ausgabe des Testprogramms sieht folgendermaßen aus (Rest der Abbildung befindet sich auf Seite 7):

```
-----
Folgende Zimmer sind noch frei:

Zimmernummer: 1
Kategorie: Doppelzimmer

Zimmernummer: 2
Kategorie: Doppelzimmer

Zimmernummer: 3
Kategorie: Doppelzimmer

Zimmernummer: 4
Kategorie: Suite

Zimmernummer: 5
Kategorie: Suite
-----

Folgende Gäste sind gespeichert:

Kundennummer: 1300
Name: Muster
Vorname: Max

Kundennummer: 1301
Name: Musterfrau
Vorname: Ilse
```

Folgende Zimmer sind noch frei:

Zimmernummer: 2

Kategorie: Doppelzimmer

Zimmernummer: 3

Kategorie: Doppelzimmer

Die Rechnung für Zimmer 5 beträgt 1750,00 €

Abbildung 3

Aufgabe 4

Themenkreis: Datenbanken

Eine Konzertagentur möchte Ihre Arbeit durch eine zentrale Datenbank unterstützen. Sie sollen eine Lösung für die Umsetzung mit einem relationalen Datenbanksystem finden. Dazu sollen Sie die notwendigen Entitäten und ihre Beziehung untereinander feststellen und in einem ER-Diagramm darstellen.

Man hat **Künstler** unter Vertrag, die **Veranstaltungen** bestreiten.

Zulieferer liefern zusätzliche **Leistungen** zu den Veranstaltungen. Dazu gibt es einen internen Leistungskatalog (Leistungsnummer und Bezeichnung). Leistungen sind z. B. Sicherheitskräfte, Schminktische, Friseure, Beleuchtung, Nebelmaschinen, Marketingmaßnahmen etc. Zu jeder Veranstaltung gibt man an, wie viele Einheiten einer Leistung man für die Veranstaltung benötigt. Ein Zuliefererverzeichnis (Zulieferernummer, jeder Zulieferer hat eine Adresse) dient der Recherche nach Zulieferern. Nach einer Veranstaltung gibt man dem Zulieferer eine Bewertung, um für die nächste Veranstaltung eine Grundlage dafür zu haben, ob man diesen Zulieferer wieder beauftragen soll. Jeder Zulieferer kann eine Leistung zu einer Veranstaltung liefern.

Es werden die Anzahl, der Preis und die Art der Bezahlung (pauschal oder pro Stück) gespeichert.

Ein **Künstler** wird durch eine Künstlernummer identifiziert. Ferner speichert man seinen Namen, seine Wunschgage und sammelt Bemerkungen über die Zusammenarbeit mit ihm.

Eine Veranstaltung findet an einer **Lokalität** an einem Datum statt, ihre Dauer (Uhrzeit von - bis) ist wichtig und der Eintrittspreis (Einheitspreis für alle).

Eine Lokalität hat eine **Adresse**, enthält die Anzahl der Plätze, die übliche Miete pro Abend und eine Art (z.B. "Hall", "OpenAir-Gelände" etc.).

Für Veranstaltungen gibt es einen Kartenvorverkauf in **Verkaufsstellen**, die Vorverkaufsgebühren in Prozent erheben. Die Gebühren hängen nur von der Verkaufsstelle, nicht aber von der Veranstaltung ab. Für jede Veranstaltung erhält jede Verkaufsstelle ein Kontingent von Karten. Nach Abschluß des Vorverkaufs sendet die Verkaufsstelle die nicht verkauften Karten zurück und in der Agentur merkt man sich, wieviele Karten pro Vorverkaufsstelle und Veranstaltung verkauft wurden. Jede Verkaufsstelle hat eine Adresse.

Adressen enthalten einen Namen (kann z. B. der Firmenname sein), einen Ansprechpartner, eine Straße (inkl. Hausnummer), ein Länderkennzeichen, Postleitzahl und Ortsname.

- a. Entwerfen Sie ein ER-Diagramm. Achten Sie auf die Kardinalitäten und die Kopplungen.
- b. Geben Sie zu jeder Entität die entsprechenden Attribute an.
- c. Kennzeichnen Sie die Primärschlüsselattribute.
- d. Welche Kriterien muss ein Schlüsselattribut erfüllen?
- e. Wie setzt sich der Schlüssel bei einer n:m Beziehung zwischen zwei Entitäten zusammen?
- f. Was versteht man unter einer Löschanomalie?
- g. Nennen Sie die Normalisierungsregeln für die ersten drei Normalformen!