
Prüfungszeitraum SoSe 2016

Studienleistung Objektorientierte Programmierung - Teil 2

Name: _____

Matr.-Nr. _____

Regularien

Bitte melden Sie sich zur Studienleistung (SL) **5152** in der EPV an.

Die Studienleistung besteht aus insgesamt drei Teilen, die im Verlauf des Semesters ausgegeben und bearbeitet werden sollen. Die Lösungen sind jeweils spätestens zu den genannten Terminen abzugeben.

	Ausgabe	Abgabe
Teil 1	Mi 20.4.2016	Mi 4.5.2016
Teil 2	Mi 4.5.2016	Mi 25.5.2016
Teil 3	Mi 25.5.2016	Mi 8.6.2016

Die Lösungen sind als .7z Datei per Mail an m.maiwald@ostfalia.de und b-u.rogalla@ostfalia.de zu senden. Im Dateinamen stehen ihr Nachname und ihre Matrikelnummer! (Nachname-Matr-Teil1.7z)

Abzugeben sind eingescannte, handschriftliche Algorithmen, UML Diagramme usw., aus denen der Programmwurf hervorgeht und die die zugrundeliegenden Ideen der Implementierung verdeutlichen.

Falls Sie für die Erstellung der UML Diagramme ein Programm verwenden (Draw.io, Visio o.ä.) sind Pdf-Ausgaben beizufügen. Zusätzlich sind der gesamte Quellcode und die Projektdateien der Visual Studio Projekte beizufügen, so dass die Programme getestet werden können.

Die Lösungen werden zeitnah von den Tutoren durchgesehen und es gibt ein Feedback zu ihren Ergebnissen.

Die Studienleistung ist bestanden, wenn Sie etwa 70 % der Aufgaben im Wesentlichen fehlerfrei bearbeitet haben.

Aufgabe 1 - Firma

Erstellen Sie eine Anwendung mit der in einer Firma die Mitarbeiter, Abteilungen und Kunden verwaltet werden können. Im Einzelnen:

- Die Mitarbeiter sind Teil der Firma und haben einen Namen, eine Anschrift, eine Personalnummer (fortlaufende Nummerierung) und beziehen ein Gehalt.
Der Name und die Personalnummer sollen von außen lesend zugreifbar sein. Die Anschrift und das Gehalt sollen von außen lesend und schreibend zugreifbar sein.
- Eine Abteilung ist Teil der Firma, hat eine Bezeichnung und besteht aus mehreren Mitarbeitern und einem Manager.
- Ein Vertreter ist ein Mitarbeiter, der für eine Vertriebsregion verantwortlich ist.
- Ein Manager ist ein Mitarbeiter, der neben seinem Gehalt einem Bonus bezieht und der eine Abteilung leitet. Der Bonus soll von außen lesend und schreibend zugreifbar sein. Der Manager „kennt“ seine Abteilung und die Abteilung „kennt“ den Manager.
- Die Kunden haben einen Namen, eine Anschrift und eine Kundennummer (fortlaufende Nummerierung). Die Kundennummer und der Rabatt sollen von außen lesend und schreibend zugreifbar sein.
- Ein Stammkunde bekommt Rabatt. Der Rabatt soll von außen lesend und schreibend zugreifbar sein.

Nutzen Sie auf sinnvolle Art die Konzepte der Vererbung. Achten Sie beim Entwurf Ihrer Klassen auf eine möglichst große Wiederverwendung. Erstellen Sie Methoden mit denen jeweils Objekte in der Verwaltung der Firma hinzugefügt und gelöscht werden können.

Dokumentieren Sie ihren Programmentwurf mit UML Diagrammen.

Testen Sie die Anwendung mit einer ausreichenden Anzahl von Objekten und Testfällen. Alle Testdaten sollen im Testprogramm vorgegeben werden. Testergebnisse sollen in nachvollziehbarer und verständlicher Form am Bildschirm ausgegeben werden.

2. Aufgabe - Lehrveranstaltung

Schreiben Sie den Kern eines Programms zur Anmeldung von Studierenden zu Lehrveranstaltungen. Aus Sicht eines Anwenders sollen folgende Vorgänge unterstützt werden:

- Anlegen einer neuen Lehrveranstaltung mit allen notwendigen Daten.
- Auflistung aller angebotenen Lehrveranstaltungen jeweils mit folgenden Daten:
 - eindeutige LVA-Nummer und Titel der Lehrveranstaltung
 - Zeitraum mit frühesten und spätesten Termin für die An- und Abmeldung
 - Anzahl der bisher angemeldeten Studierenden
- An- und Abmeldung einer/eines Studierenden zu einer Lehrveranstaltung innerhalb des definierten Zeitraums. Eine Studentin bzw. ein Student kann zu einer Lehrveranstaltung nur einmal angemeldet sein.
- Auflistung aller zu einer Lehrveranstaltung angemeldeten Studierenden mit Matrikelnummern und Namen.
- Löschen eines Studierenden.
- Löschen einer Lehrveranstaltung, wobei auch alle Anmeldungen von Studierenden zu dieser Veranstaltung verlorengehen.

Schreiben Sie nur den Kern eines entsprechenden Programms (ohne Benutzerschnittstelle) sowie ein Testprogramm.

Dokumentieren Sie ihren Programmentwurf mit UML Diagrammen.

Testen Sie die Anwendung mit einer ausreichenden Anzahl von Objekten und Testfällen. Alle Testdaten sollen im Testprogramm vorgegeben werden. Testergebnisse sollen in nachvollziehbarer und verständlicher Form am Bildschirm ausgegeben werden

3. Aufgabe - Seminarverwaltung

Schreiben Sie ein Programm mit der sich einzelne Personen oder mehrere Mitarbeiter einer Firma zu einer Seminarveranstaltung anmelden (buchen) können. Im Einzelnen sind folgende Anforderungen zu implementieren:

- Anlegen einer Seminarveranstaltung mit Kürzel, Titel, Seminarort (Adresse), Dozent, Kosten, Anfangsdatum und -uhrzeit, Enddatum und -uhrzeit, Teilnehmerzahl und maximale Teilnehmerzahl.
- Anlegen von Teilnehmern, d.h. von Einzelpersonen oder von Firmen, mit mehreren teilnehmenden Personen. Für jeden Teilnehmer sind die Adresse (PLZ, Ort, Straße, Nr.) zu speichern.
- Für jede Person sind Anrede, Titel, Name und Vorname, sowie die Kontaktdaten (Telefon, Mail, Social-Media) zu speichern.
- Für Firmen sind der Firmenname, der Ansprechpartner (= 1. Teilnehmer), die Kontaktdaten und die Anzahl der Teilnehmer zu speichern, nicht aber die Daten jedes teilnehmenden Mitarbeiters.
- Nach der Anmeldung erhält jeder Teilnehmer (Person oder Firma) eine Anmeldebestätigung, die ein Datum, eine Buchungsnummer und einen Freitext enthält. Außerdem wird an jedem Teilnehmer (Person oder Firma) eine Rechnung mit Rechnungsnummer, Rechnungsdatum und Rechnungsbetrag mit ausgewiesener MwSt. versandt.

Verwenden Sie Klassen um zusammenhängende Daten jeweils zu kapseln (z.B. Adressen oder Kontaktdaten...). Falls möglich setzen Sie auch Aggregation, Komposition oder Vererbung ein.

Dokumentieren Sie ihren Programmentwurf mit UML Diagrammen.

Testen Sie die Anwendung mit einer ausreichenden Anzahl von Objekten und Testfällen. Alle Testdaten sollen im Testprogramm vorgegeben werden. Testergebnisse sollen in nachvollziehbarer und verständlicher Form am Bildschirm ausgegeben werden.