

Software Engineering Implementation 1

Semesterprojekt

www.technikerschule-hf.ch
Herbstsemester 2013

Aufgabe 4: **Objektorientiertes Design & Programmieren (OOA/D/P)** **(Entwurfsmodell erstellen und implementieren)**

Ausgabetermin: 05.10.2013

Abgabetermin: 28.10.2013

Ausgangslage:

Auf Basis des vorliegenden Analysemodells müssen Sie unter besonderer Beachtung der nicht-funktionalen Anforderungen für die 1. Iteration ein Entwurfsmodell erstellen und dieses als Prototyp in der Programmiersprache Java implementieren. Die 1. Iteration umfasst die Klasse (bzw. Klassen) Ihres Analysemodells, die die Abstraktion "Kunde" modelliert. Für die 1. Iteration realisieren Sie die Normalabläufe der Anwendungsfälle "Kunde erstellen" und "Kunde anzeigen". Es genügt, wenn Sie dabei die Eigenschaften Vor- und Nachname von Kunden berücksichtigen.

Auftrag:

1. Ergänzen Sie das Glossar um die ins Englische übersetzten Namen der Klassen des Fachklassenmodells.
2. Überführen Sie für die 1. Iteration das Fachklassenmodell des Analysemodells in ein Fachklassenmodell für das Entwurfsmodell. Methoden sowie mögliche Assoziations- und Vererbungsbeziehungen sind anzugeben, Attribute nicht. Stellen Sie das angepasste Entwurfsmodell mit einem UML Klassendiagramm dar.
3. Integrieren Sie das Fachklassenmodell in die gegebene logische 3-Schichtenarchitektur (siehe Seite 3).
4. Überprüfen Sie mit einem UML Sequenzdiagramm Ihr Entwurfsmodell für den Normalablauf des Anwendungsfalles „Kunde erstellen“ hinsichtlich Vollständigkeit und Verantwortlichkeiten der Klassen sowie Interaktionen der entsprechenden Objekte. Ergänzen Sie bei Bedarf das Entwurfsmodell nach dieser Überprüfung.
5. Implementieren Sie für die 1. Iteration den Teil Ihres Entwurfsmodells, der es ermöglicht einen Kunden mit seinem Vor- und Nachnamen zu speichern. Implementieren Sie dazu einen Prototypen in Java.
6. Testen Sie Ihren Prototypen für die Normalabläufe der Anwendungsfälle "Kunde erstellen" und "Kunde anzeigen", indem Sie mehrere Kunden erstellen, speichern, lesen und auf der Konsole deren Vor- und Nachnamen ausgeben.
7. Dokumentieren Sie den Java-Quelltext (nur Dateiköpfe und Methoden) mittels Javadoc und erzeugen Sie die zugehörigen HTML-Dokumentationsseiten.

Hinweis: Beachten Sie die Hinweise auf Seite 4.

Software Engineering Implementation 1

Semesterprojekt

www.technikerschule-hf.ch
Herbstsemester 2013

Bewertung (mittels Kontrolle und/oder Gespräch, max. 36 Punkte):

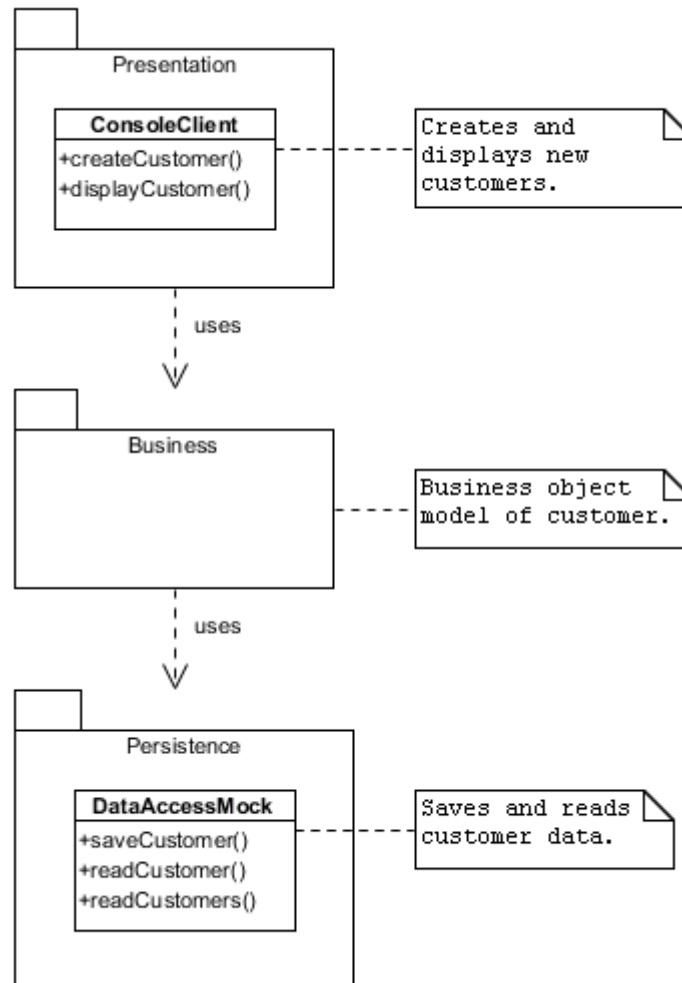
#	Bewertungskriterien (Punkte: 0 => nicht erfüllt; 1 => <50% erfüllt; 2 => >=50% erfüllt; 3 => 100% erfüllt)	Punkte			
		0	1	2	3
1	Das UML Klassendiagramm zum Fachklassenmodell ist fachlich korrekt und vollständig.				
2	Die Integrations des Fachklassenmodells in die gegebene logische 3-Schichtenarchitektur ist fachlich korrekt und vollständig.				
3	Das UML Sequenzdiagramm ist fachlich korrekt und vollständig.				
4	Die ins Englische übersetzten Namen der Klassen des Fachklassenmodells sind fachlich korrekt und vollständig ins Glossar eingetragen.				
5	Die Implementation der Persistence-Schicht (DataAccessMock) ist korrekt und vollständig.				
6	Die Implementation der Business-Schicht (Fachklassenmodell) ist korrekt und vollständig.				
7	Die Implementation der Presentation-Schicht (ConsoleClient) ist korrekt und vollständig.				
8	Der Prototyp läuft fehlerfrei (Namen der Kunden werden gespeichert und gelesen).				
9	Die Quellcode-Dokumentation mit Javadoc ist korrekt und vollständig.				
11	Namen bzw. Bezeichnungen in den UML Diagrammen sind aussagekräftig und die bekannten UML Namenskonventionen werden beachtet.				
12	Namen bzw. Bezeichnungen im Java-Quelltext sind aussagekräftig und die bekannten Java Namenskonventionen werden beachtet.				

Software Engineering Implementation 1

Semesterprojekt

www.technikerschule-hf.ch
Herbstsemester 2013

Gegebene logische 3-Schichtenarchitektur:



Software Engineering Implementation 1

Semesterprojekt

www.technikerschule-hf.ch
Herbstsemester 2013

Hinweise zu den Aufträgen 5 und 6:

- Allgemein:
 - Implementieren Sie nicht alles auf einmal, sondern gehen Sie nach dem Prinzip vor „Code a little, test a little“.
 - Implementieren Sie nur die Anwendungsfälle „Kunde anlegen“ und „Kunde anzeigen“.
 - Speichern und lesen Sie nur Vor- und Nachname der Kunden.
- Vorbereitung:
 - Erstellen Sie ein neues Eclipse-Projekt.
 - Legen Sie im neuen Eclipse-Projekt folgende Java-Packages an:
 - Presentation-Schicht
ch.<WURZEL>.persistence
 - Business-Schicht
ch.<WURZEL>.business
 - Persistence-Schicht
ch.<WURZEL>.presentation
- 1. Persistence-Schicht:
 - Implementieren Sie die Klasse DataAccessMock.
 - Speichern Sie im DataAccessMock Kundendaten (Vor- und Nachname) nicht im Dateisystem, sondern in-memory (im Hauptspeicher mittels Arrays und Strings).
- 2. Business-Schicht
 - Implementieren Sie aus Ihrem Fachklassenmodells nur die Teile, die für die Anwendungsfälle "Kunde erstellen" und "Kunde anzeigen" benötigt werden.
- 3. Presentation-Schicht
 - Implementieren Sie die Klasse ConsoleClient.
 - Erstellen und speichern Sie mehrere Kunden
 - Verzichten Sie auf Benutzereingaben über die Tastatur und arbeiten Sie stattdessen mit "harten" Testdaten für die Daten der Kunden
 - Lesen Sie die Kunden und geben Sie diese wieder auf der Konsole aus.

Hinweise zum Auftrag 7:

- Nur Dateiköpfe und Methoden dokumentieren.
- getter- und setter-Methoden dürfen zusammengefasst dokumentiert werden.