

Projektarbeit

Anforderungsanalyse

**Design eines Kundenverwaltungssystems für die Firma
PackZeugs AG**

Datum: 30.11.2013

Autoren: Pascal Kern
David Marmy

Klasse: TSI1209I

Inhaltsverzeichnis

3. Aufgabe: Objektorientierte Analyse (OOA) (Analysemodell/Fachklassenmodell erstellen)

3.1.Fachklassenmodell zu Anwendungsfälle	4
Anwendungsfall „Kunde Erstellen“ und „Kunde anzeigen“	4
3.2.Sequenzdiagramm für Normalablauf	4
Anwendungsfall „Kunde Erstellen“	4
3.3.Zustandsdiagramm für die Abstraktion „Kunde“	5
3.4.Glossar zur Fachliche Verantwortlichkeit	5

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

4.1.Glossar ergänzen englische Namen der Klassen	6
Bemerkung zu den folgenden Diagrammen	7
4.2.Fachklassendiagramm 1. Iteration in Entwurfsmodell Fachklassenmodell überführen	7
4.3.Fachklassenmodell in 3-Schichtenarchitektur integrieren	8
4.4.Entwurfsmodell mit Sequenzdiagramm überprüfen	9
4.5.Entwurfsmodell in Prototyp implementieren	9
4.6.Prototyp testen	10
4.7.Quelltext per Javadoc als HTML-Dokument erstellen	10

5. Aufgabe: Objektorientierte Analyse, Design & Programmieren (OOA/D/P) (Analyse- und Entwurfsmodell anpassen, Entwurfsmodell implementieren)

5.1.Anpassungen (Fach-) Klassendiagramm - Analysemodell	11
5.2.Angepasstes Klassendiagramm - (Entwurf-) Designmodell	12
5.3.Erweiterung Design- Entwurfsmodell in “Business” und “Persistence” Schicht um abstrakten Klassen und Schnittstellen	13
5.4.Klassen- und Komponentendiagramm mit “Singleton” und “Factory” (GoF) Design Pattern auf Business und Persistenc Schicht (3-Schichten Modell)	13
5.5.Sequenzdiagramm zur Überprüfung Normalfall “Kunde erstellen” - Korrekturen ableiten	13
5.6.Glossar ergänzen um Klassen und Schnittstellen aus Design- Entwurfsmodell [5.3]	13
5.7.Angepasstes Desing- Entwurfsmodell implementieren [5.3]	13
5.8.JUnit Test auf “Business” und “Persistenz” Schicht für Normalfall “Kunde erstellen”	13
5.9.Normalfall “Kunde erstellen” und “Kunde anzeigen” testen	13
5.10.JavaDoc generieren	13

6. Anhang

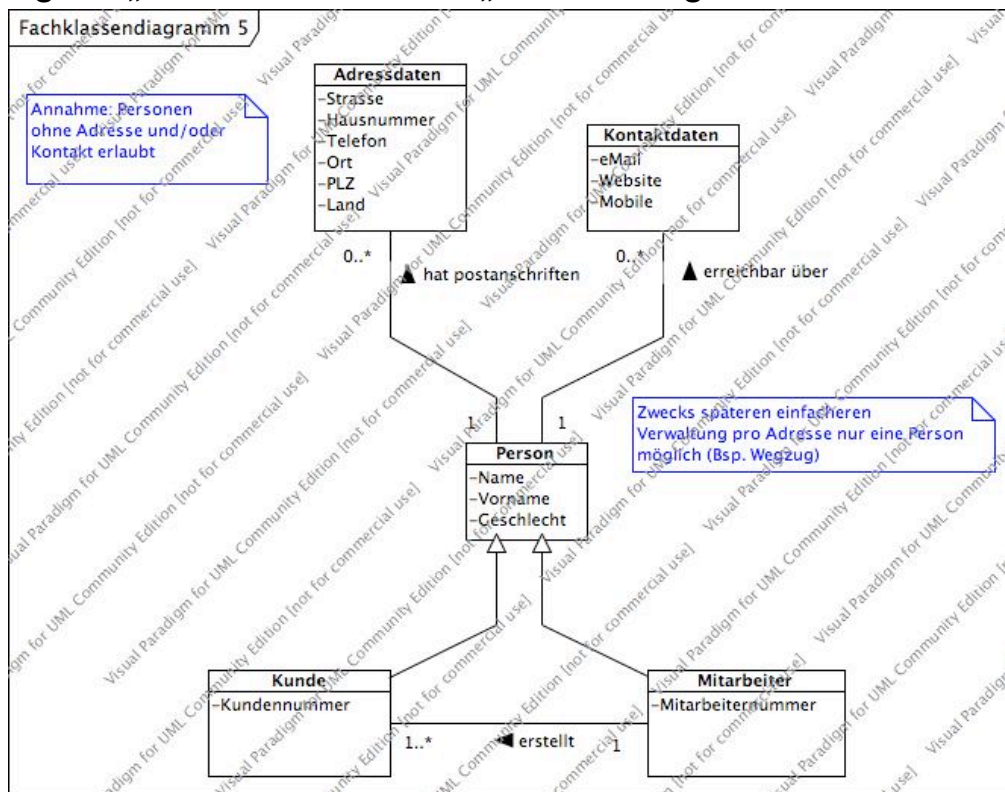
6.1.Aufgabenstellungen	14
Aufgabenstellung 3:	14
Bewertungskriterien Aufgabe 3:	14
Bewertung Aufgabe 3 (Teil 1,2 und 4) eMail vom 07.10.2013:	15
Bewertung Aufgabe 3 (Teil 3) eMail vom 30.10.13:	15
Aufgabenstellung 4:	16
Bewertungskriterien Aufgabe 4:	17
Bewertung Aufgabe 4 eMail vom 27.11.13	17
Aufgabenstellung 5:	18
Bewertungskriterien Aufgabe 5:	19
6.2.Abgaberichtlinien	20
6.3.Zusatzinfos zu den Aufgaben	21
Aufgabe 4	21
Aufgabe 5	23

Inhaltsverzeichnis

3. Aufgabe: Objektorientierte Analyse (OOA) (Analysemodell/Fachklassenmodell erstellen)

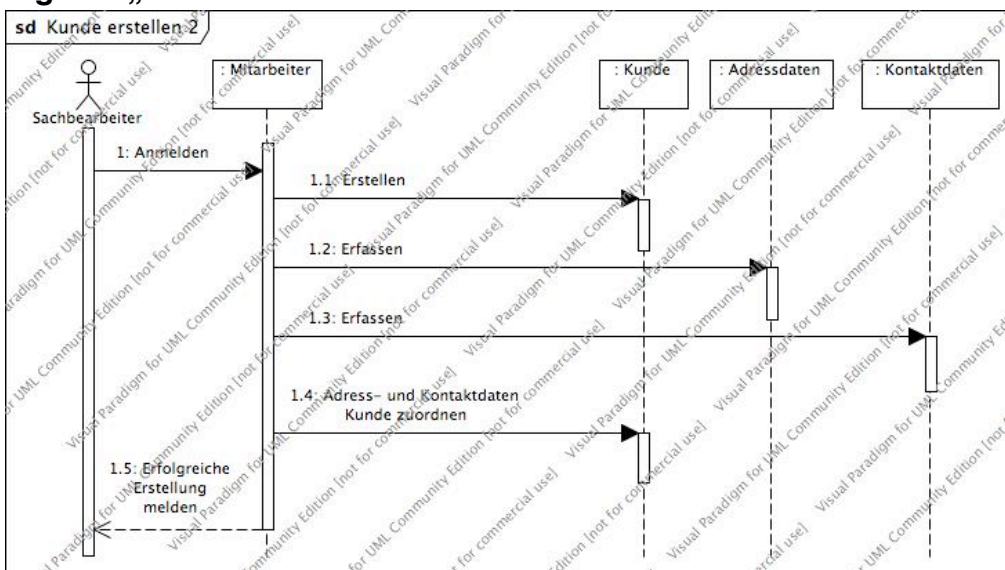
3.1. Fachklassenmodell zu Anwendungsfälle

Anwendungsfall „Kunde Erstellen“ und „Kunde anzeigen“

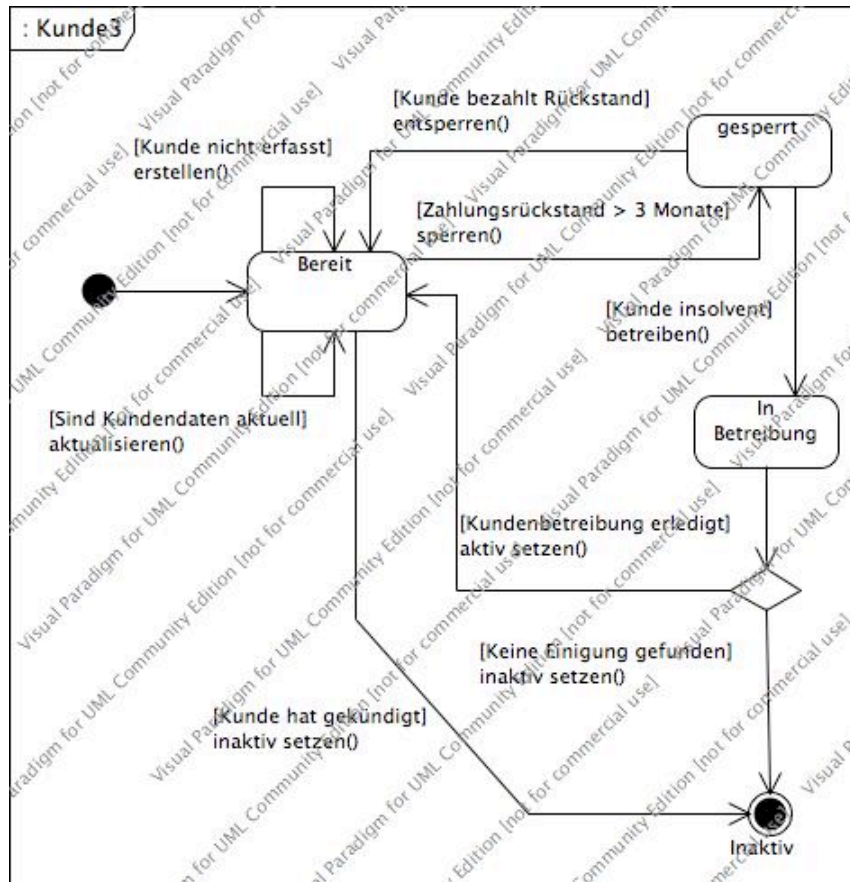


3.2. Sequenzdiagramm für Normalablauf

Anwendungsfall „Kunde Erstellen“



3.3. Zustandsdiagramm für die Abstraktion „Kunde“



3.4. Glossar zur Fachliche Verantwortlichkeit

Bezeichnung	Fachliche Verantwortung der Klasse	Synonyme (nicht bewerten)
Kunde	Identifikation eines Bedarfsträgers beinhaltet Informationen zur Geschäftsbeziehung zwischen ihm (dem Kunden) und der Unternehmung. Bezieht gegen ein entsprechendes Entgelt eine Leistung vom (unserem) Unternehmen.	Klient, Konsument, Verbraucher
Person	<u>Abstrakte Darstellung!</u> Stellt den Zugang zu zusätzlichen Daten wie Anschrift, Kontaktinformationen eines (Geschäfts-) Partner zur Verfügung. Beispiel: Gemeinsamkeiten von Kunden, Mitarbeitern aber kann auch für Lieferanten oder Geschäftspartner stehen.	Leute, Mensch
Mitarbeiter	Identifiziert jemand, der für die Unternehmung mit anderen zusammenarbeitet; Daten pflegt und Änderungen im Kundensystem vornimmt.	Personal, Belegschaft, Angestellte(r)
Adressdaten	Trägt Informationen zur Bezeichnung des Wohnorts, der Straße und der Hausnummer einer schriftlichen Kontaktmöglichkeit. Postalische Domizil-Daten einer Person oder Unternehmung.	Anschrift, Kontaktstelle
Kontakt	Daten für die direkte Kommunikation mit einer Person. Ergänzend zu den Adressdaten wie allfällige Telefon, eMail, Website, Ansprechperson usw.	Koordinaten, Kontakt-Personalien

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

4.1. Glossar ergänzen englische Namen der Klassen

Der Vollständigkeit halber haben wir neben den Namen und Synonymen auch die Bedeutung übersetzt. Dies muss nicht unbedingt bewertet werden.

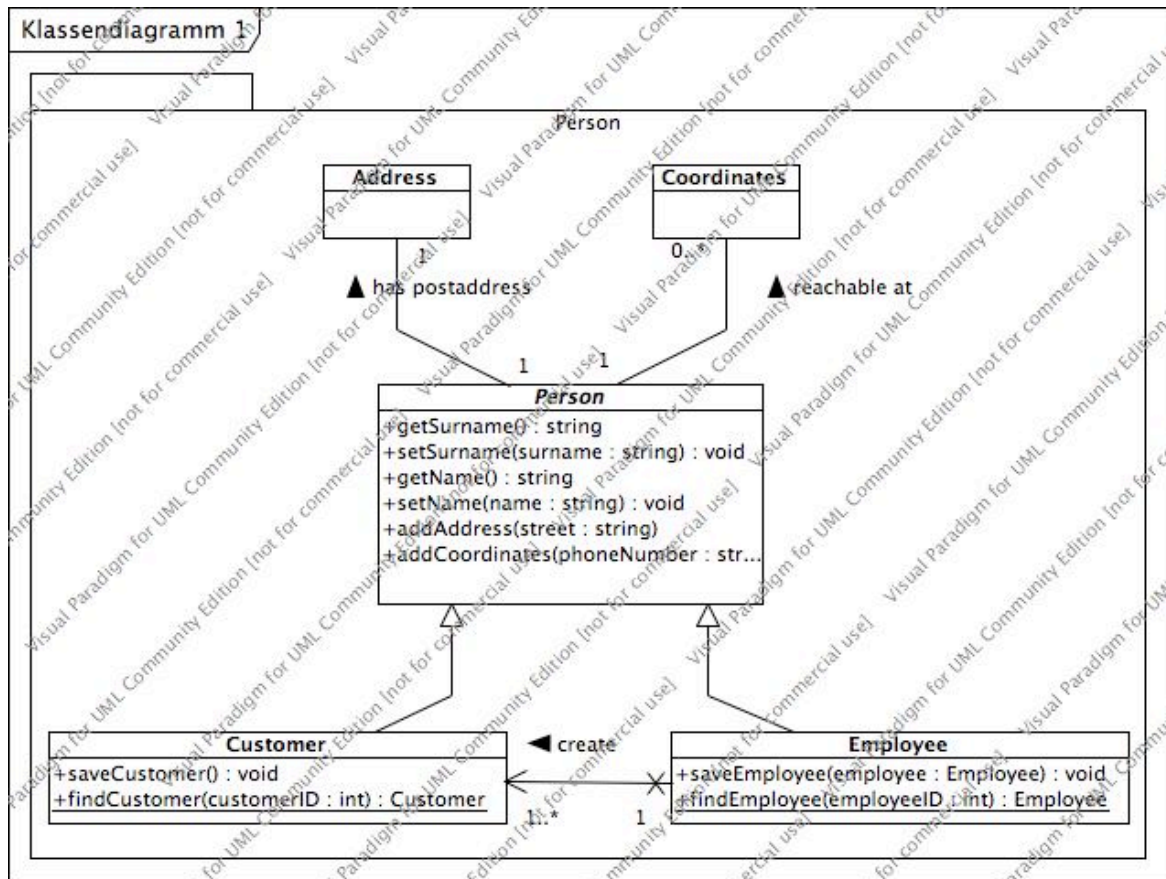
Bezeichnung	Übersetzung	Synonyme (nicht bewerten)	Übersetzung	Beschreibung Description
Kunde	Customer	Klient, Konsument, Verbraucher	Client, Consumer, --	Identifikation eines Bedarfsträgers beinhaltet Informationen zur Geschäftsbeziehung zwischen ihm (dem Kunden) und der Unternehmung. Bezieht gegen ein entsprechendes Entgelt eine Leistung vom (unserem) Unternehmen. Identification of a carrier includes information required for business relationship between him (the client) and the company. Refers to rates of pay, a performance from (our) company.
Person	Person	Leute, Mensch	People, Human	<u>Abstrakte Darstellung!</u> Stellt den Zugang zu zusätzlichen Daten wie Anschrift, Kontaktinformationen eines (Geschäfts-) Partner zur Verfügung. Beispiel: Gemeinsamkeiten von Kunden, Mitarbeitern aber kann auch für Lieferanten oder Geschäftspartner stehen. <u>Abstract representation!</u> Provides access to additional information such as address, contact information of a (business) partner. For example, similarities of customers, employees but can also stand for suppliers or business partners.
Gender	Geschlecht			Angabe zum Kunden ob er männlich oder weiblich ist. Informations about the client if it's a male or female.
Mitarbeiter	Employee	Personal, Belegschaft, Angestellte(r)	Personnel, Workforce, Employee(s)	Identifiziert jemand, der für die Unternehmung mit anderen zusammenarbeitet; Daten pflegt und Änderungen im Kundensystem vornimmt. Identifies someone who works for the company with others, maintains and modifies data in the customer system.
Adressdaten	Address	Anschrift, Kontaktstelle	--, Point of contact	Trägt Informationen zur Bezeichnung des Wohnorts, der Straße und der Hausnummer einer schriftlichen Kontaktmöglichkeit. Postalische Domizil-Daten einer Person oder Unternehmung. Carries information about the name of the place, the street and the house number written contact information. Postal domicile of a person or company data.
Kontakt	Coordinates	Koordinaten, Kontakt-Personalien	Coordinates, Contact-Personnel	Daten für die direkte Kommunikation mit einer Person. Ergänzend zu den Adressdaten wie allfällige Telefon, eMail, Website, Ansprechperson usw. Data for direct communication with a person. In addition to the address as any telephone, email, website, contact person, etc.

Bemerkung zu den folgenden Diagrammen

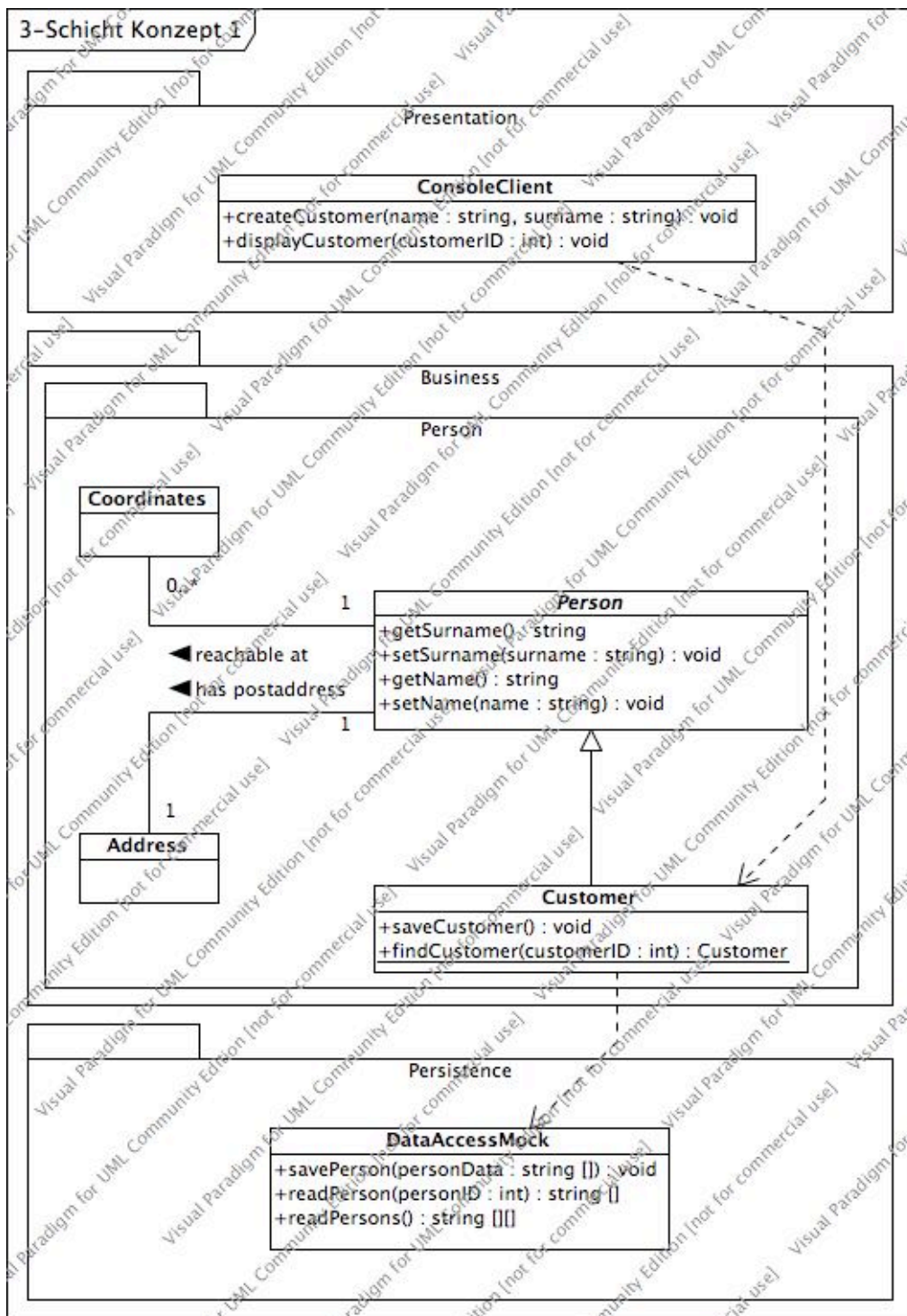
Gemäss Aufgabenstellung wurden nur die Eigenschaften Vor- und Nachname aufgeführt. Die im Analysemodell vorhandenen Eigenschaften Geschlecht (gender), Kundennummer (clientID) und Mitarbeiternummer (staffID) wurden also nicht "vergessen" sondern bewusst weg gelassen.

Bei den Methoden wurden die für die Klassen Adresse (Adress) und Kontakt (Coordinates) aber dennoch aufgenommen um die Logik einigermassen komplett dar zu stellen.

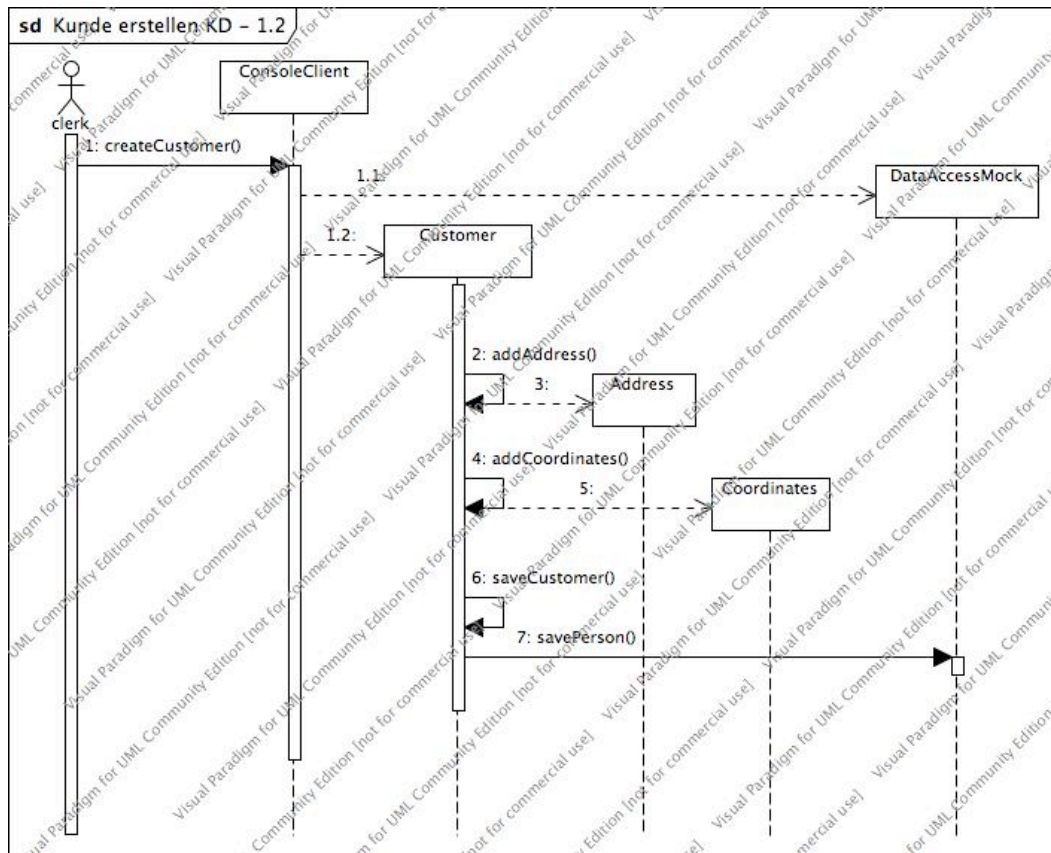
4.2. Fachklassendiagramm 1. Iteration in Entwurfsmodell Fachklassenmodell überführen



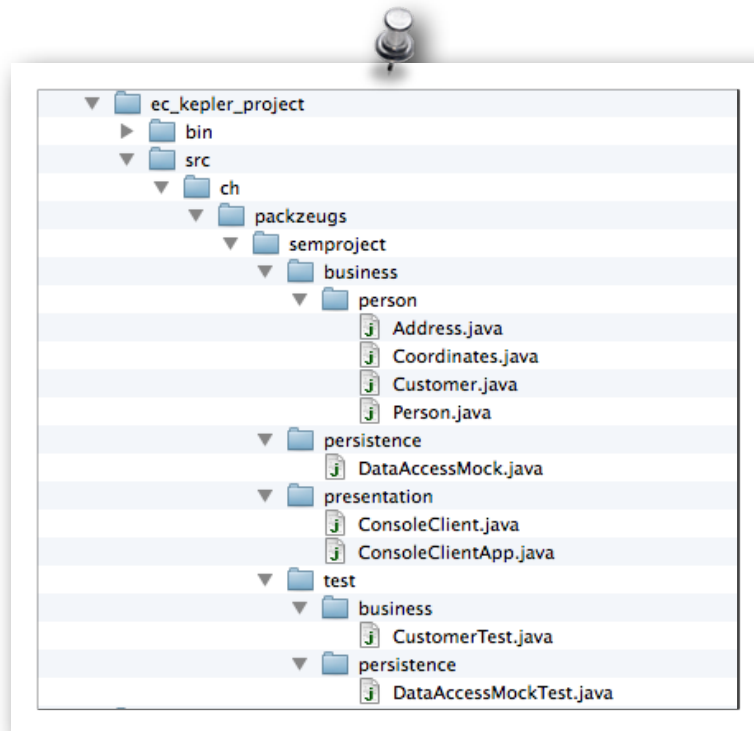
4.3. Fachklassenmodell in 3-Schichtenarchitektur integrieren



4.4. Entwurfsmodell mit Sequenzdiagramm überprüfen



4.5. Entwurfsmodell in Prototyp implementieren

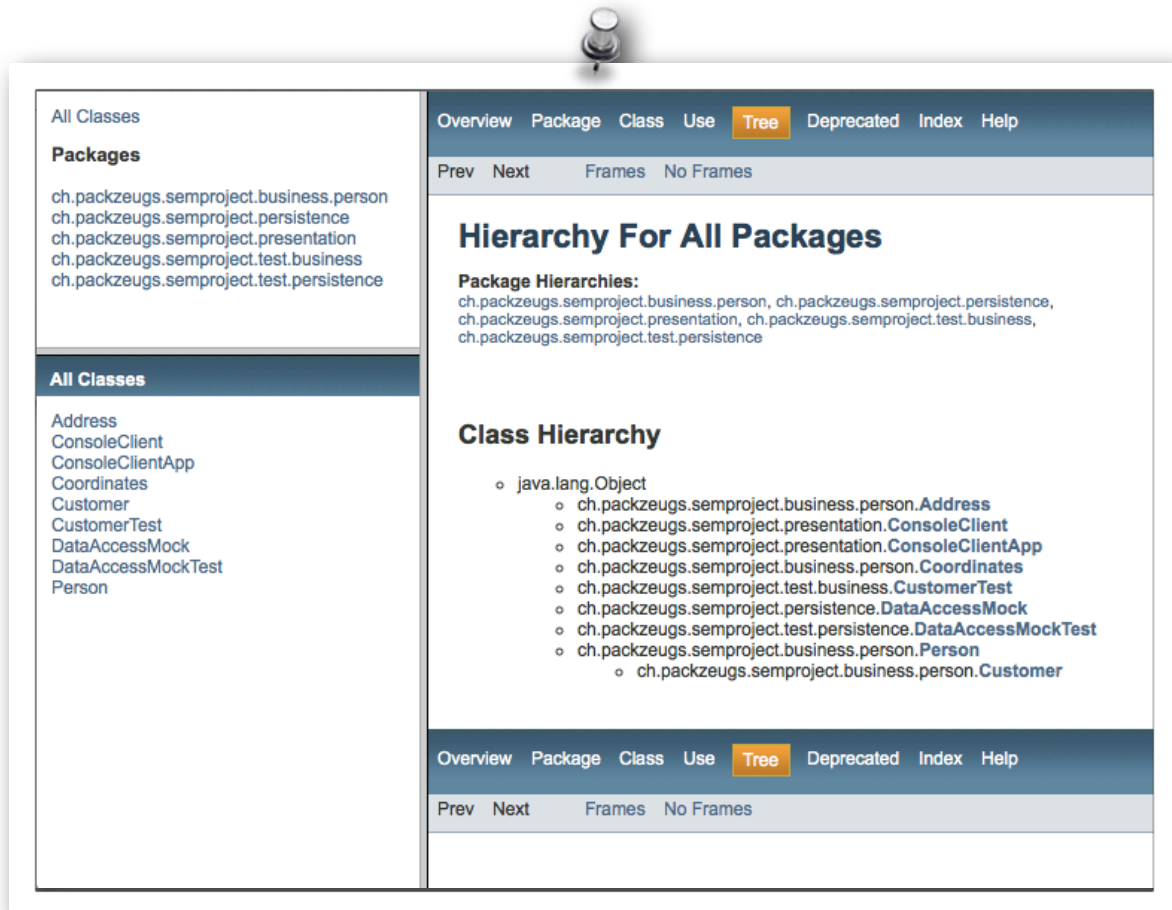


4.6. Prototyp testen

Einzelne Pakete (Schichten vom 3-Schicht Konzept) einzeln erfolgreich mit den Tests im Paket "ch.packzeugs.semproject.test" getestet.

Das Zusammenspiel der drei Schichten wurde mit der Klasse "ConsoleClientApp" erfolgreich durchgeführt und kann somit auch als erfolgreich getestet betrachtet werden.

4.7. Quelltext per Javadoc als HTML-Dokument erstellen



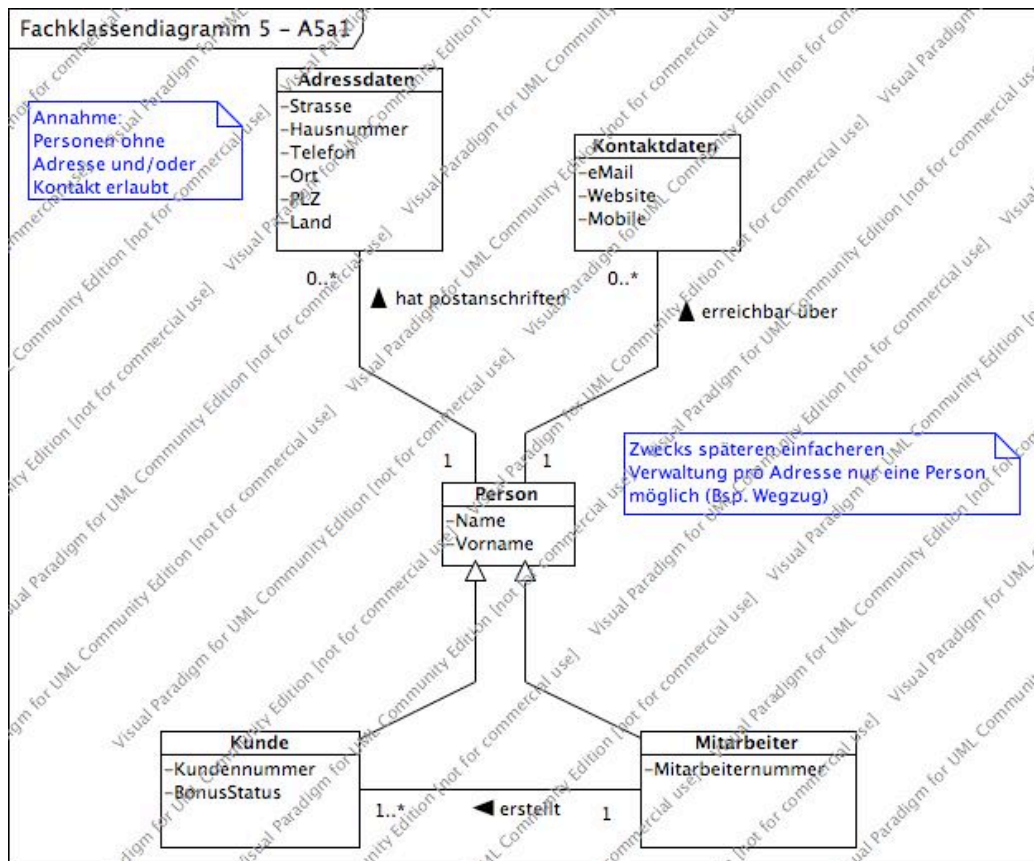
The screenshot displays a Javadoc-generated HTML document. On the left, there is a sidebar with two sections: "All Classes" and "Packages". The "All Classes" section lists several classes: Address, ConsoleClient, ConsoleClientApp, Coordinates, Customer, CustomerTest, DataAccessMock, DataAccessMockTest, and Person. The "Packages" section lists several package names: ch.packzeugs.semproject.business.person, ch.packzeugs.semproject.persistence, ch.packzeugs.semproject.presentation, ch.packzeugs.semproject.test.business, and ch.packzeugs.semproject.test.persistence.

The main content area is titled "Hierarchy For All Packages" and contains a "Package Hierarchies" section. Below this, there is a "Class Hierarchy" section. The class hierarchy is a tree structure starting from java.lang.Object. The classes listed in the hierarchy are: ch.packzeugs.semproject.business.person.Address, ch.packzeugs.semproject.presentation.ConsoleClient, ch.packzeugs.semproject.presentation.ConsoleClientApp, ch.packzeugs.semproject.business.person.Coordinates, ch.packzeugs.semproject.test.business.CustomerTest, ch.packzeugs.semproject.persistence.DataAccessMock, ch.packzeugs.semproject.test.persistence.DataAccessMockTest, ch.packzeugs.semproject.business.person.Person, and ch.packzeugs.semproject.business.person.Customer.

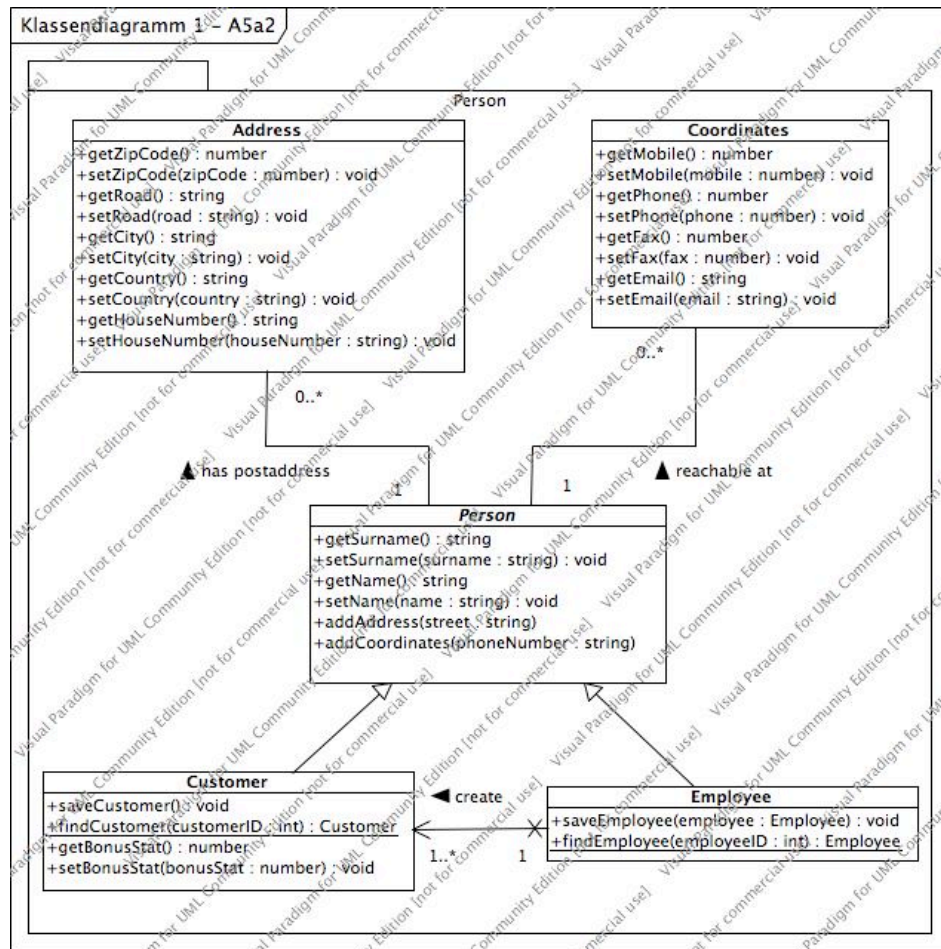
The document has a navigation bar at the top with links: Overview, Package, Class, Use, Tree (selected), Deprecated, Index, and Help. There are also links for Prev, Next, Frames, and No Frames.

5. Aufgabe: Objektorientierte Analyse, Design & Programmieren (OOA/D/P) (Analyse- und Entwurfsmodell anpassen, Entwurfsmodell implementieren)

5.1. Anpassungen (Fach-) Klassendiagramm - Analysemodell



5.2. Angepasstes Klassendiagramm - (Entwurf-) Designmodell



- 5.3. Erweiterung Design- Entwurfsmodell in “Business” und “Persistence” Schicht um abstrakten Klassen und Schnittstellen**
- 5.4. Klassen- und Komponentendiagramm mit “Singleton” und “Factory” (GoF) Design Pattern auf Business und Persistence Schicht (3-Schichten Modell)**
- 5.5. Sequenzdiagramm zur Überprüfung Normalfall “Kunde erstellen” - Korrekturen ableiten**
- 5.6. Glossar ergänzen um Klassen und Schnittstellen aus Design- Entwurfsmodell [5.3]**
- 5.7. Angepasstes Design- Entwurfsmodell implementieren [5.3]**
- 5.8. JUnit Test auf “Business” und “Persistence” Schicht für Normalfall “Kunde erstellen”**
- 5.9. Normalfall “Kunde erstellen” und “Kunde anzeigen” testen**
- 5.10. JavaDoc generieren**

A. Anhang

1. Aufgabenstellungen

Aufgabenstellung 3:

Aufgabe 3: Objektorientierte Analyse (OOA) (Analysemodell/Fachklassenmodell erstellen)

Ausgabetermin: 17.09.2013

Abgabetermin: 30.09.2013

Ausgangslage:

Die Anforderungen an das Kundenverwaltungssystem liegen systematisch dokumentiert vor. Aus den Anforderungen sollen Sie nun ein objektorientiertes Analysemodell herleiten, dass in UML und textuell zu beschreiben ist. Dabei ist die Kundensprache Deutsch zu verwenden.

Auftrag:

1. Erstellen Sie ein Fachklassenmodell, dass die Realisierung der Anwendungsfälle "Kunde erstellen" und "Kunde anzeigen" ermöglicht. Das Fachklassenmodell muss aus mindestens zwei Klassen bestehen. Attribute sowie mögliche Assoziations- und Vererbungsbeziehungen sind anzugeben, Methoden nicht.
2. Überprüfen Sie Ihr Fachklassenmodell mittels eines UML Sequenzdiagramms für den Normalablauf des Anwendungsfalls „Kunde erstellen“ hinsichtlich Vollständigkeit, Verantwortlichkeiten der Klassen und Interaktionen der entsprechenden Objekte. Ergänzen Sie bei Bedarf das Fachklassenmodell nach dieser Überprüfung.
3. Überprüfen Sie Ihr Fachklassenmodell mittels eines UML Zustandsdiagramms für die Abstraktion "Kunde". Ergänzen Sie bei Bedarf das Fachklassenmodell nach dieser Überprüfung.
4. Beschreiben Sie die fachliche Verantwortlichkeit jeder Klasse in einem Glossar
Hinweise:
a) Objektorientierte Konzepte und/oder UML Diagramme sind nicht zu erläutern
b) Informationen aus dem UML Klassendiagramm sind nicht zu wiederholen

Bewertungskriterien Aufgabe 3:

Bewertung (mittels Kontrolle und/oder Gespräch, max. 18 Punkte):

#	Bewertungskriterien (Punkte: 0 => nicht erfüllt; 1 => <50% erfüllt; 2 => >50% erfüllt; 3 => 100% erfüllt)	Punkte			
		0	1	2	3
1	Die Klassen im UML Klassendiagramm sind fachlich korrekt und vollständig				
2	Assoziations- und/oder Vererbungsbeziehungen im UML Klassendiagramm sind fachlich korrekt und vollständig				
3	Das UML Sequenzdiagramm ist fachlich korrekt und vollständig				
4	Das UML Zustandsdiagramm ist fachlich korrekt und vollständig				
5	Die Verantwortlichkeiten der Klassen sind fachlich korrekt und vollständig im Glossar beschrieben				
6	Namen bzw. Bezeichnungen sind aussagekräftig und die bekannten UML Namenskonventionen werden beachtet				

Bewertung Aufgabe 3 (Teil 1,2 und 4) eMail vom 07.10.2013:

Hallo zusammen,

bei der 3. Aufgabe (ohne Auftrag 3) des Semesterprojekts habt Ihr 12 Punkte erreicht.

Die Punkte sind wie folgt von links nach rechts auf die in der Aufgabe angegebenen Bewertungskriterien von oben nach unten zu verteilen (1. von links auf 1. von oben, 2. links auf 2. von oben usw.):

2 3 2 0 3 2

Beste Grüsse
Arif

Zwischenergebnis:

12 Punkt von max 15 = 5.0

Bewertung Aufgabe 3 (Teil 3) eMail vom 30.10.13:

Hallo zusammen,

bei der 3. Aufgabe (jetzt mitAuftrag 3) des Semesterprojekts habt Ihr 14 Punkte erreicht.

Die Punkte sind wie folgt von links nach rechts auf die in der Aufgabe angegebenen Bewertungskriterien von oben nach unten zu verteilen (1. von links auf 1. von oben, 2. links auf 2. von oben usw.):

2 3 2 2 3 2

Beste Grüsse
Arif

Endergebnis:

14 Punkt von max 18 = 4.88

Aufgabenstellung 4:**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 Anwendungsfalls „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.

Bewertungskriterien Aufgabe 4:**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.				

Bewertung Aufgabe 4 eMail vom 27.11.13

Hallo zusammen,

Ihr habt 30 Punkte erreicht.

Die Punkte sind wie folgt von links nach rechts auf die in der Aufgabe angegebenen Bewertungskriterien von oben nach unten zu verteilen (1. von links auf 1. von oben, 2. links auf 2. von oben usw.):

2 3 2 3 3 3 3 3 2 3 3

Beste Grüsse

Arif

Aufgabenstellung 5:

Aufgabe 5:
Objektorientierte Analyse, Design & Programmieren (OOA/D/P)
(Analyse- und Entwurfsmodell anpassen, Entwurfsmodell implementieren)

Ausgabetermin: 30.11.2013

Abgabetermin: 10.12.2013, 02.01.2014

Ausgangslage:

Folgende neuen Anforderungen der Firma PackZeug AG liegen für die 2. Iteration vor:

- Kunden können mehr als eine Anschrift, ein Telefon, eine E-Mail etc. haben
- Je nach Anzahl Bestellungen im Jahr haben Kunden einen Bronze-, Silber- oder Goldstatus. Der Status bringt den Kunden Vorteile wie z.B. Preisrabatte

Die 2. Iteration umfasst folgende Aufgaben:

- Umsetzung der neuen Anforderungen
- Vollständige Behandlung der Kundendaten (Name, Vorname und Kontaktinformationen)
- Automatisiertes Testen der Module
- Umsetzung der nicht-funktionalen Anforderungen Wartbarkeit und Wiederverwendbarkeit

Im Rahmen dieser Aufgaben müssen Sie das vorliegende Analyse- und Entwurfsmodell anpassen und das angepasste Entwurfsmodell in der Programmiersprache Java implementieren.

Aufträge:

1. Passen Sie das UML Klassendiagramm (Fachklassenmodell) des Analysemodells an für die neuen Anforderungen.
2. Passen Sie das UML Klassendiagramm des Entwurfsmodells an das angepasste UML Klassendiagramm des Analysemodells an.
3. Erweitern Sie Ihr Entwurfsmodell auf den Schichten "Business" und "Persistence" um abstrakte Klassen und Schnittstellen. Stellen Sie das angepasste Entwurfsmodell mit einem UML Klassen- und Komponentendiagramm dar.
4. Wenden Sie auf den Schichten "Business" und "Persistence" die Entwurfsmuster "Singleton" und "Factory" an. Stellen Sie das angepasste Entwurfsmodell mit einem UML Klassen- und Komponentendiagramm dar.
5. Überprüfen Sie mit einem UML Sequenzdiagramm Ihr Entwurfsmodell für den Normalablauf des Anwendungsfalls „Kunde erstellen“ hinsichtlich Vollständigkeit und Verantwortlichkeiten der Klassen und Schnittstellen sowie Interaktionen der entsprechenden Objekte. Ergänzen Sie bei Bedarf das Entwurfsmodell nach dieser Überprüfung.
6. Ergänzen Sie das Glossar um die neuen abstrakten Klassen sowie Schnittstellen aus dem Entwurfsmodell.
7. Implementieren Sie Ihr angepasstes Entwurfsmodell.
8. Implementieren Sie für die Schichten "Business" und "Persistence" Unit-Tests mit JUnit für den Anwendungsfall „Kunde erstellen“.
9. Testen Sie Ihre Implementierung für die Normalabläufe der Anwendungsfälle "Kunde erstellen" und "Kunde anzeigen", indem Sie mehrere Kunden erstellen, speichern, lesen und auf der Konsole deren Daten ausgeben.
10. Dokumentieren Sie den Java-Quelltext (nur Dateiköpfe und Methoden) mittels Javadoc und erzeugen Sie die zugehörigen HTML-Dokumentationsseiten.

Bewertungskriterien Aufgabe 5:**Bewertung (mittels Kontrolle und/oder Gespräch, max. 45 Punkte):**

#	Bewertungskriterien (Punkte: 0 => nicht erfüllt; 1 => >0% & <50% erfüllt; 2 => >=50% erfüllt; 3 => 100% erfüllt)	Punkte			
		0	1	2	3
1	Die Erweiterung des Fachklassenmodell (UML Klassendiagramm) des Analysemodells um die neuen Anforderungen ist korrekt und vollständig.				
2	Das Entwurfsmodell (UML Klassendiagramm) auf Basis des angepassten Analysemodells ist korrekt und vollständig.				
3	Die Erweiterung des Entwurfsmodells um abstrakte Klassen und Schnittstellen ist korrekt und vollständig.				
4	Die Anwendung der Entwurfsmuster "Singleton" und "Factory" auf das Entwurfsmodell ist korrekt und vollständig.				
5	Szenario (UML Sequenzdiagramm) für das Entwurfsmodell für den ausgewählten Anwendungsfall ist korrekt und vollständig.				
6	Abstrakte Klassen sowie Schnittstellen aus dem Entwurfsmodell sind korrekt und vollständig ins Glossar eingetragen.				
7	Implementation der Persistenzschicht (DataAccessMock) ist korrekt und vollständig.				
8	Implementation der Geschäftsschicht (fachliches Entwurfsmodell) ist korrekt und vollständig.				
9	Implementation der Präsentationsschicht (ConsoleClient) ist korrekt und vollständig.				
10	Die Implementation der Unit-Tests für die Persistence-Schicht ist korrekt und vollständig.				
11	Die Implementation der Unit-Tests für die Business-Schicht ist korrekt und vollständig.				
12	Die Implementation läuft fehlerfrei (Daten der Kunden werden gespeichert und gelesen).				
13	Quelltext-Dokumentation mit Javadoc ist korrekt und vollständig.				
14	Namen in den UML Diagrammen sind aussagekräftig und die bekannten UML Namenskonventionen werden beachtet.				
15	Namen im Java-Quelltext sind aussagekräftig und die bekannten Java Namenskonventionen werden beachtet.				

2. Abgaberichtlinien

Die Abgabe der bearbeiteten Projektteilaufgaben erfolgt per E-Mail an den Dozenten. Folgender Betreff ist für die E-Mail zu verwenden:

- SE AD: Team <#>: Aufgabe <#>
Beispiel: SE AD: Team 3: Aufgabe 3
- SE AD: <Vorname> <Nachname>: Aufgabe <#>
Beispiel: SE AD: Beat Müller: Aufgabe 2

Erlaubte Dateiformate

- PDF
- ODF-Formate (ODT, ODS)
- JPG
- ZIP
- Microsoft-Formate (DOCX, XLSX)

Regeln für Dateinamen

Dateien sind nach folgendem Schema zu benennen:

- aufgabe_<#>_team_<#>.<erweiterung>
Beispiel: aufgabe_1_team_3.odt
- aufgabe_<#>_<vorname>_<nachname>.<erweiterung>
Beispiel: aufgabe_1_beat_mueller.odt

Hinweis: Keine Leer- oder Sonderzeichen für Dateinamen verwenden

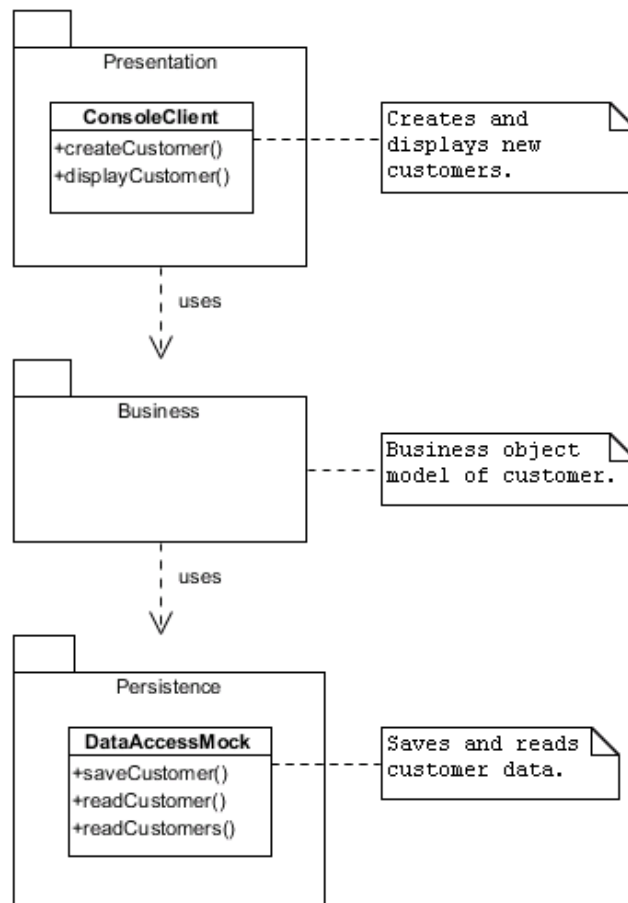
Dokumentenform

- Die Lösungsteile (Texte, Tabellen, Diagramme etc.) zu einer Aufgabe sind in einem einzelnen Dokument zusammenzufassen
- Ein Dokument für alle Aufgaben ist zu verwenden
- Das Titelblatt des Dokuments muss folgende Angaben beinhalten
 - Fach/Vorlesungstitel: Technikerschule HF Zürich - Software Engineering (A&D)
 - Haupttitel Projektarbeit, Untertitel gemäss Aufgabe
 - Datum (TT.MM.JJJJ)
 - Vorname(n) und Nachname(n) des Autors (der Autoren)
 - Klasse
- Je Aufgabe ein Hauptkapitel mit einem entsprechenden Namen
- UML-Diagramme sind als Grafiken in das Dokument zu integrieren. Falls dies nicht vernünftig darstellbar möglich ist, sind die betreffenden UML-Diagramme als JPG-Dateien beizulegen und im Dokument entsprechend zu referenzieren
- Texte, Tabellen, Diagramme etc. sind durch Überschriften bzw. Titeln den einzelnen Aufträgen einer Aufgabe zuzuordnen

3. Zusatzinfos zu den Aufgaben

Aufgabe 4

Gegebene logische 3-Schichtenarchitektur:



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.

Aufgabe 5

Hinweise:

- 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.
- Auftrag 7:
 - Optional können Sie Array durch ArrayList ersetzen, um die Implementierung zu vereinfachen
- Auftrag 8:
 - Legen Sie für die Unit-Tests folgende neuen Java-Packages an:
 - Business-Schicht
ch.<WURZEL>.test.business
 - Persistence-Schicht
ch.<WURZEL>.test.presentation
- Auftrag 10:
 - Nur Dateiköpfe und Methoden dokumentieren
 - getter- und setter-Methoden dürfen zusammengefasst dokumentiert werden