12. Erste Schritte in der Objektorientierte Analyse mit CRC-Karten

Prof. Dr. rer. nat. Uwe Aßmann
Lehrstuhl Softwaretechnologie (ST)
Institut für Software- und Multimediatechnik (SMT)
Fakultät für Informatik
TU Dresden
Version 22-0.1, 30.04.22

1) CRC-Analyse
2) CRC mit Gruppen
3) CRRC-Analyse

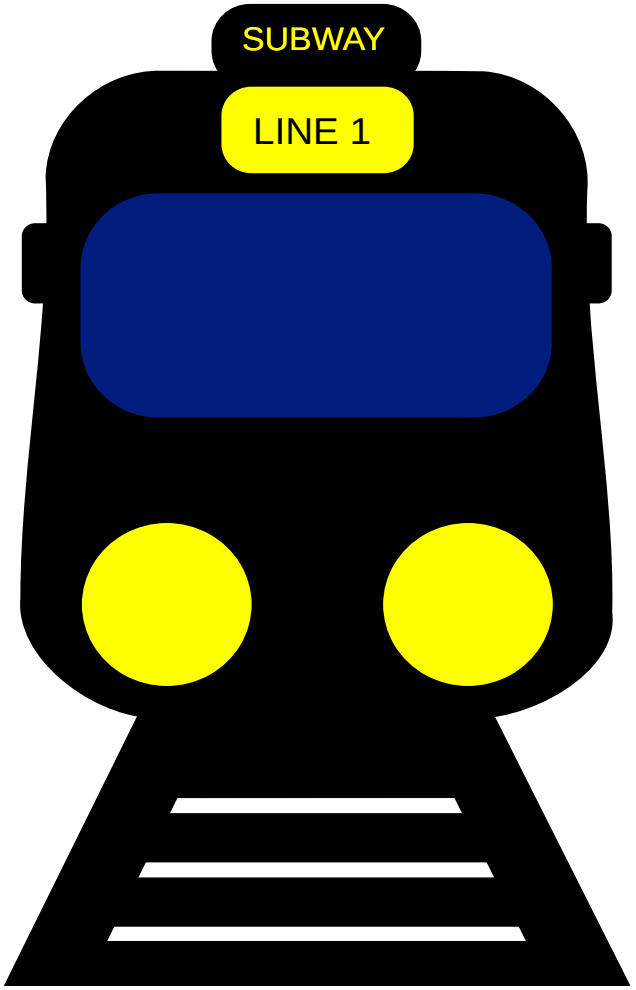
Bitte selbständig studieren!
Begleitmaterial zur Analyse
Wird in Kap 31 noch mal aufgenommen

Literatur

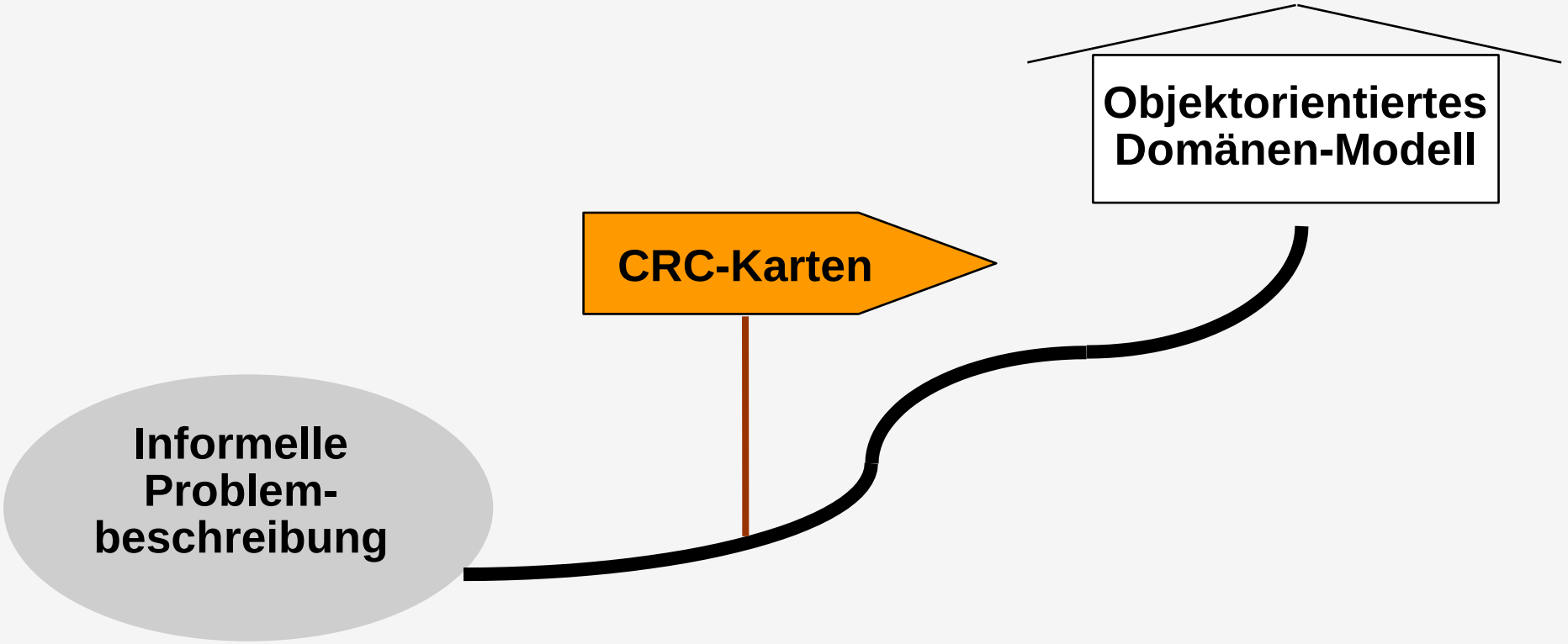
- Obligatorische Literatur
  - Zuser Kap 9
  - Beck, Kent; Cunningham, Ward (October 1989), "A laboratory for teaching object oriented thinking", ACM SIGPLAN Notices (New York, NY, USA: ACM) 24 (10): 1–6, http://c2.com/doc/oopsla89/paper.html
- Weiterführende Literatur
  - HotDraw CRC cards http://c2.com/doc/crc/draw.html
  - Scott Ambler. The Object Primer. Cambridge University Press. Gutes Kapitel über CRC
  - [PP] Andrew Hunt, David Thomas. The pragmatic programmer. Addison-Wesley. Deutsch: Der Pragmatische Programmierer. Hanser-Verlag. Leseprobe
    - http://www.beck-shop.de/fachbuch/leseprobe/9783446223097_Excerpt_004.pdf

Q 0.1 Java Herunterladen

- Das Java Development Kit (JDK)
- https://adoptopenjdk.net/, brew tap adoptopenjdk/openjdk
- http://openjdk.java.net/, brew info openjdk

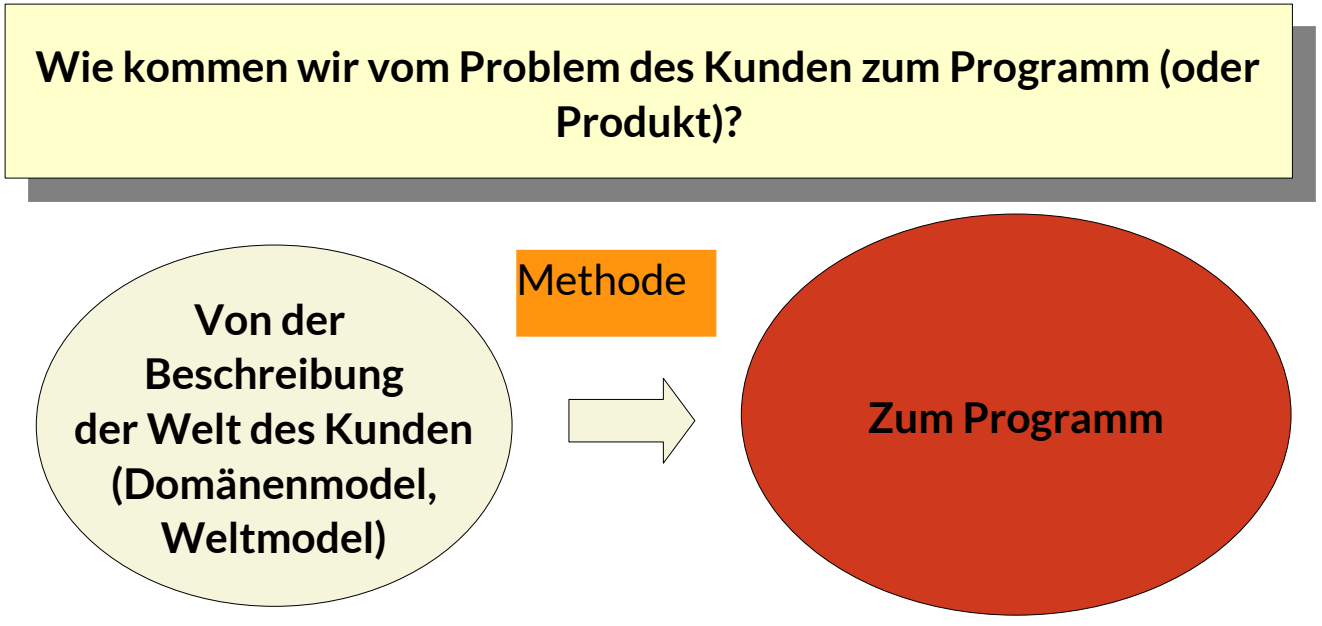


12.1 Analysemethoden: Analyse mit CRC-Karten



Die Rolle von Methoden in der Softwareentwicklung

- Um ein System zu entwickeln, sollte man sich an eine ***Softwareentwicklungs-Methode*** halten, die einem durch alle Schritte leitet
  - Eine Methode beruht auf einer oder mehreren zentralen Fragen, die immer wieder gestellt werden
  - Analysemethode – Entwurfsmethode - Implementierungsmethode



Objektorientierte Entwurfsmethodik

- Die die Entwicklung treibenden Fragen sind:

Welche Objekte (Klassen) enthält ein System?
Welche Verantwortlichkeiten haben sie?

- Vom objektorientierten Entwurf existieren einige Spielarten, die zusätzliche Hilfsmittel einsetzen.
- CRC-Karten
- Strukturgetriebener Klassenentwurf (z.B. nach Balzert, später)

Analyse mit CRC-Karten

- ***CRC*** = **C**lass – **R**esponsibility – **C**ollaborator
  (Klasse – Verantwortlichkeit – Mithelfer)
- [Beck, Cunningham, Wilkerson, Wirfs-Brock (ca. 1989-1995)]
- Technik zur Gruppenarbeit (Rollenspiele)
- Wichtigstes Hilfsmittel: Zu beschriftende Karteikarten

(Vorderseite)

Klassenname (class)	
Ober- und Unterklassen (opt.)	
Verantwort-lichkeiten **(responsibili-ties)**	**Mithelfer** **(collabora-tors)**

(Rückseite)

Klassenname (class)
Definition
Attribute **(attributes)**

Class Responsibility Cards (CRC)

Welches Objekt ist für welche Aufgaben zuständig?

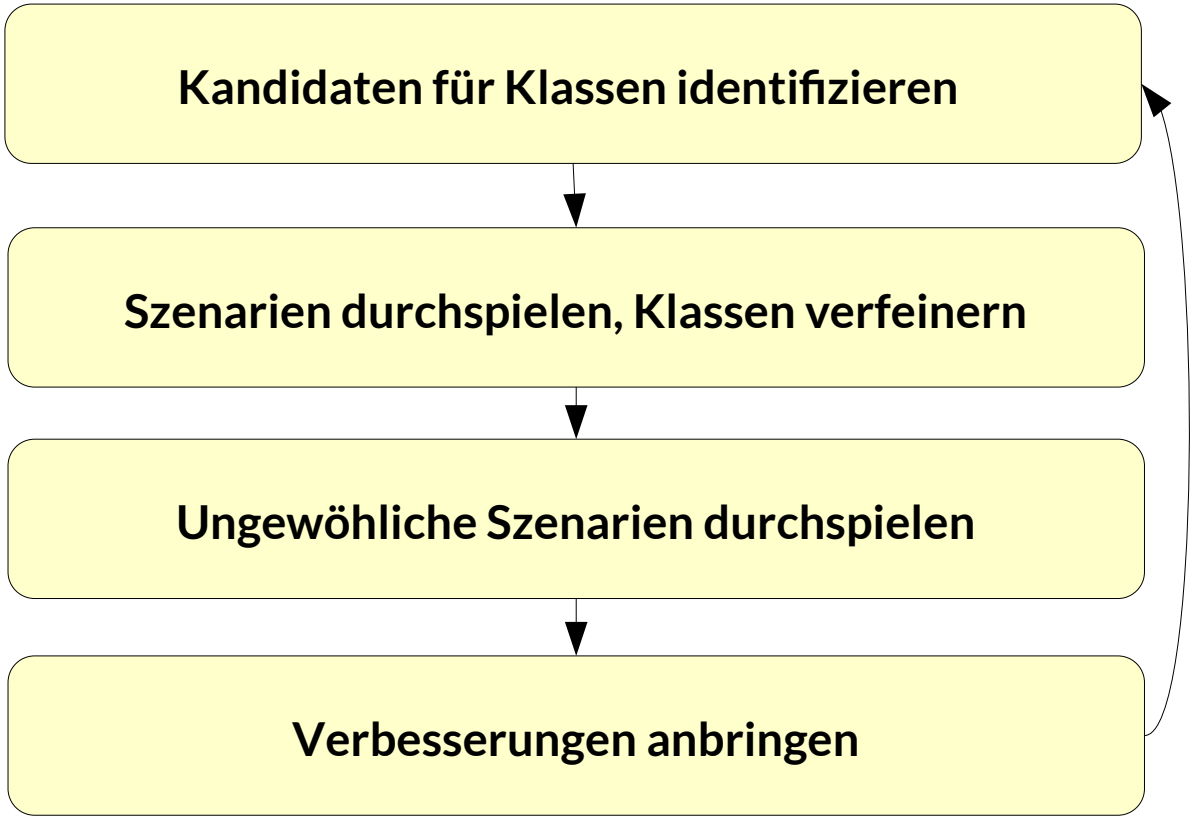
- Zuständigkeit für Aufgabe (Dienst):
  - Aktion
  - Auskunft (query)
  - Prüfung (check)
- Kooperation mit Partner
  - Wenn Klasse alleine zur Bewältigung der Aufgaben nicht fähig
  - “Mit wem muss ich kooperieren, um einen Dienst zu erhalten?”

(Vorderseite)

Klassenname (class)	
Ober- und Unterklassen	
Verantwort-lichkeiten **(responsibili-ties)**	**Mithelfer** **(collabora-tors)**

CRC-Karten-Methode: Vorgehensweise als Einzelner

- Voraussetzung: informelle Anforderungsbeschreibung (ideal: ausführliche Anforderungsspezifikation)



Gesetz 51 [PP]: Nicht Anforderungen sammeln, sondern nach ihnen schürfen

Szenarien

- Ein **Szenario** ist ein typischer Ablauf von Aktionen zur Erfüllung des geplanten Systemzwecks.
  - z.B. notwendige Schritte zur Einrichtung einer neuen Teambesprechung:
    - Festlegung des Titels
    - Festlegung der Teilnehmer
    - Festlegung eines ersten Terminvorschlags und Abstimmung
    - Festlegung eines Besprechungsraums
  - Viele Szenarien zur Beschreibung eines Teilaspekts des Systemzwecks.
- Szenarien werden interaktiv "durchgespielt". Man stellt sich die Entwicklungsfragen der CRC-Methode:
  - Wer ist zuständig (Verantwortungsbereich)?
  - Welche Aufgaben sind dazu zu erfüllen? (auf Karte eintragen)
  - Welches Privatwissen ist dazu nötig? (auf Karte eintragen)

Identifikation von Klassen-Kandidaten: Substantiv-Verb-Analyse

- Analysiere textuelle Anforderungen:
  - Finde *Objekte* aus Hauptwörtern
  - Finde *Kooperationen* aus Subjekt-Objekt-Relationen, Genitiven, Nebensätzen
  - Finde *Aktivitäten* aus Verben und substantivierten Verben
  - Finde *Kontextklassen* durch Orte und adverbiale Bestimmungen

Finde Aufgaben aus Verben, Klassen aus Hauptwörtern

- Beispiel
- "When the student orders a computer, he has to contact the computer dealer. The computer dealer ships the packet *via the parcel service*."
- “When the driver turns on the lights the battery is discharged. When the engine runs the dynamo recharges the battery...”

Kriterien für Klassen in der Substantiv-Verb-Analyse

- **Problemrelevante Substantive** auswählen
  - Allgemeine Worte weglassen (z.B. "System")
  - Auch "versteckte" Substantive betrachten (z.B. "Privattermin")

- **Konkretisierung:** Hat jede Klasse einen klar abgegrenzten Verantwortungsbereich?
  - Gibt es Aufgaben, die spezifisch für die Objekte der Klasse sind?
  - Passen die Aufgaben zusammen?
  - Gibt es "Privatwissen", das ein Objekt der Klasse besitzt?

- **Allokation**: Sind Verantwortungsbereiche von mehreren Klassen abgedeckt?

- **Vollständigkeit**: "Haben wir alles?"
  - Gibt es nicht als Substantive erwähnte wichtige "Mitspieler"?

Klassen finden: Beispiel Terminverwaltung

Problembeschreibung (user story):

Es ist ein Terminverwaltungssystem für Arbeitsgruppen zu entwickeln.

Das System soll alle geplanten Teambesprechungen (z.B. Projektbesprechungen) speichern und die Reservierung von Besprechungsräumen unterstützen.

Das System soll automatisch Kollisionen mit bereits bekannten Terminen vermeiden. Deshalb soll auch die Eintragung privater Termine möglich sein.

Beispiel einer CRC-Karte

Teambesprechung	
Oberklassen: Termin *Unterklassen:*	
Titel wissen Datum wissen Teilnehmer wissen Teilnehmer einladen Raum festlegen	Teammitglied Besprechungsraum

Rückseite:

Teambesprechung
Ein Objekt 'Teambesprechung' beschreibt genau einen Termin, an dem mehrere Teilnehmer der Gruppe teilnehmen sollen.
Titel Datum

12.2. CRC in Gruppen

CRC-Karten-Methode: Vorgehensweise in Gruppe

- Das Kernstück der Methode sind intensive Gruppensitzungen.
- **Voraussetzung**: informelle, textuelle Anforderungsbeschreibung (ideal: ausführliche Anforderungsspezifikation)
  - Kandidaten für Klassen (Karten) identifizieren
  - Karten auf einem Tisch oder Whiteboard plazieren.
    - Kollaborierende Karten nah zueinander plazieren, andere voneinander entfernen. Anordnung ständig ändern, je mehr Zusammenarbeiten zustandekommen
    - "Heisse" Karten in die Mitte des Tisches
- **Durchführung**
  - Typische Szenarien identifizieren und durchspielen (dabei: Karten schrittweise ausfüllen)
  - Iteration: Verbesserungen, mehrfache Wiederholung
  - Ungewöhnliche Szenarien durchspielen

Gesetz 17 [PP]: Programmieren Sie nahe am Problem (an der Sprache der Anwender)

Die Rollen im Gruppenspiel

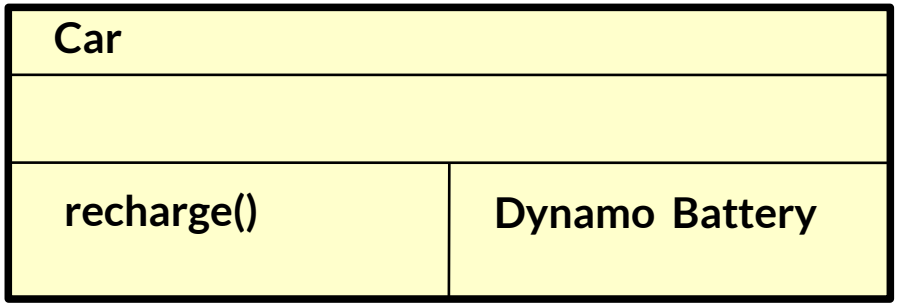
- Ideale Gruppengröße: 5 bis 6 aktive TeilnehmerInnen
- **Rollen der Teilnehmer**:
  - Fachspezialisten, ev. Kunden
  - Systemanalytiker (Anforderungsanalyst)
  - Systementwickler
  - Manager (?)
  - Moderator, 'Facilitator'

- **Gruppendynamik**:
  - CRC-Karten-Sitzungen können Teamgeist stärken
  - Vorhandene Gruppen-Probleme können aufbrechen
  - **Kein** Mittel zur Klärung und Lösung von Problemen im Team !

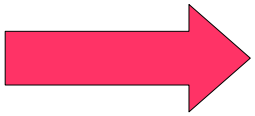
Gesetz 52 [PP]: Arbeiten Sie mit Anwendern zusammen, damit Sie denken wie ein Anwender

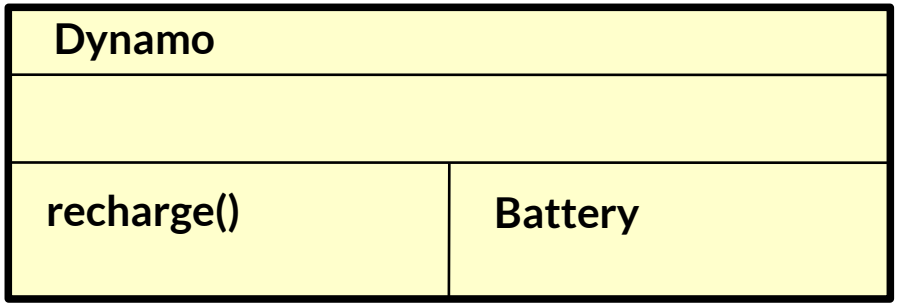
Vorsicht: Klassen des Kontexts

- Oft werden Verantwortlichkeiten fälschlicherweise einer Klasse zugeordnet, die eigentlich eine Klasse im *Kontext* ist
- Achte auf Ortsbestimmungen und Urheberschaften:
  - "the car recharges the battery *via the dynamo*"
- Achte auf Passivsätze

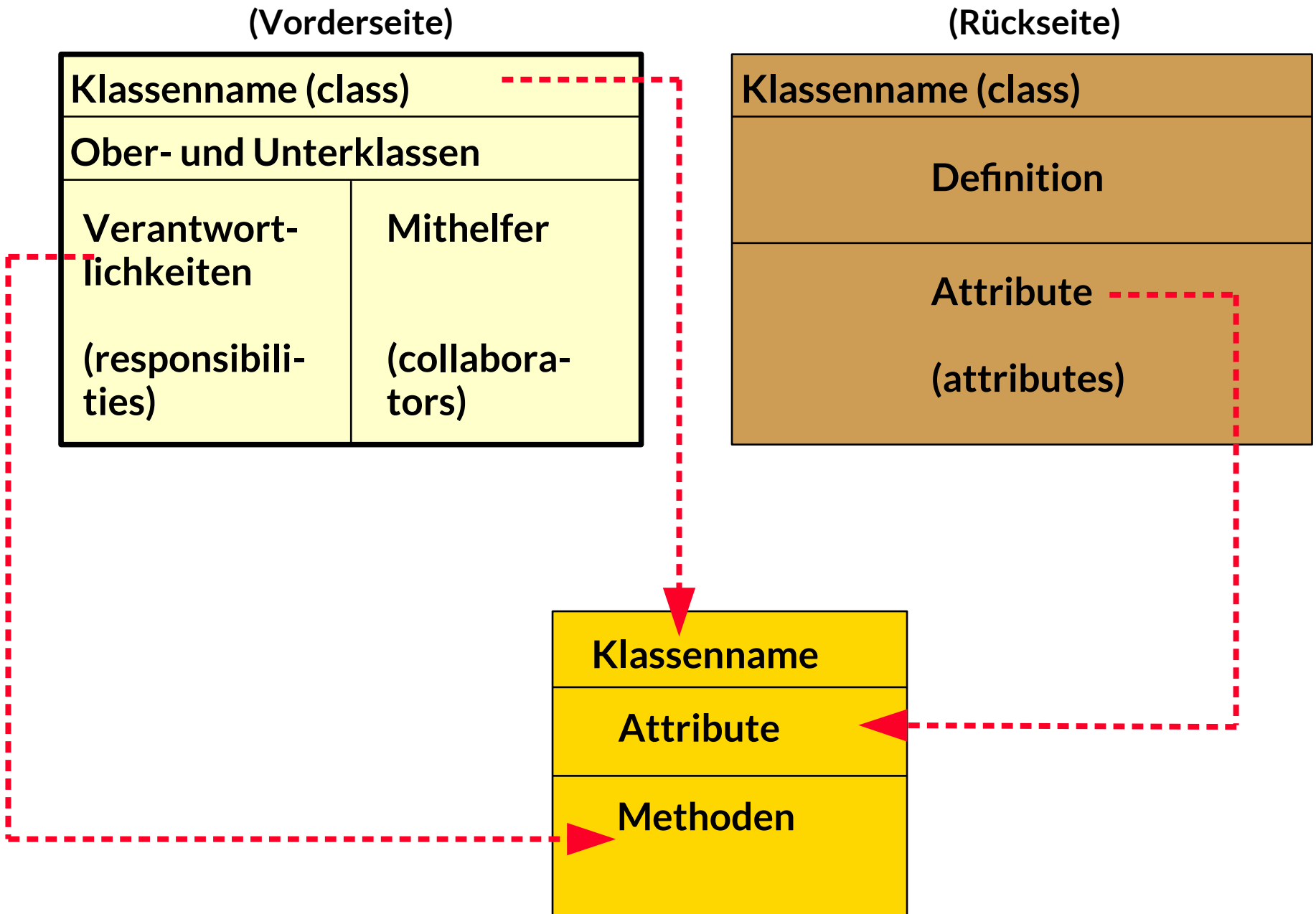


Car ist eine Klasse im *Kontext* von Dynamo

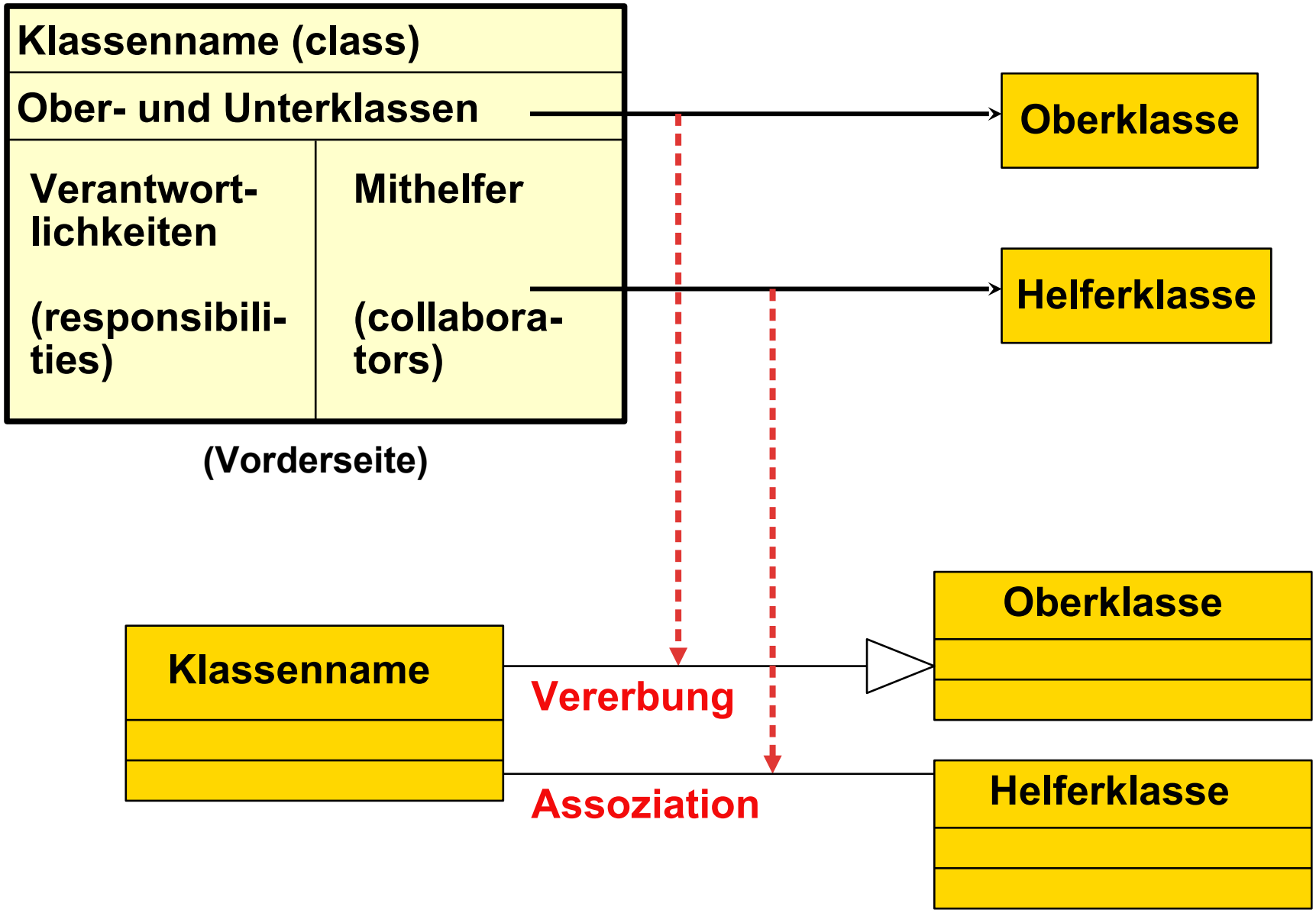




Von CRC-Karten zum UML-Klassenmodell (1)



Von CRC-Karten zum UML-Klassenmodell (2)



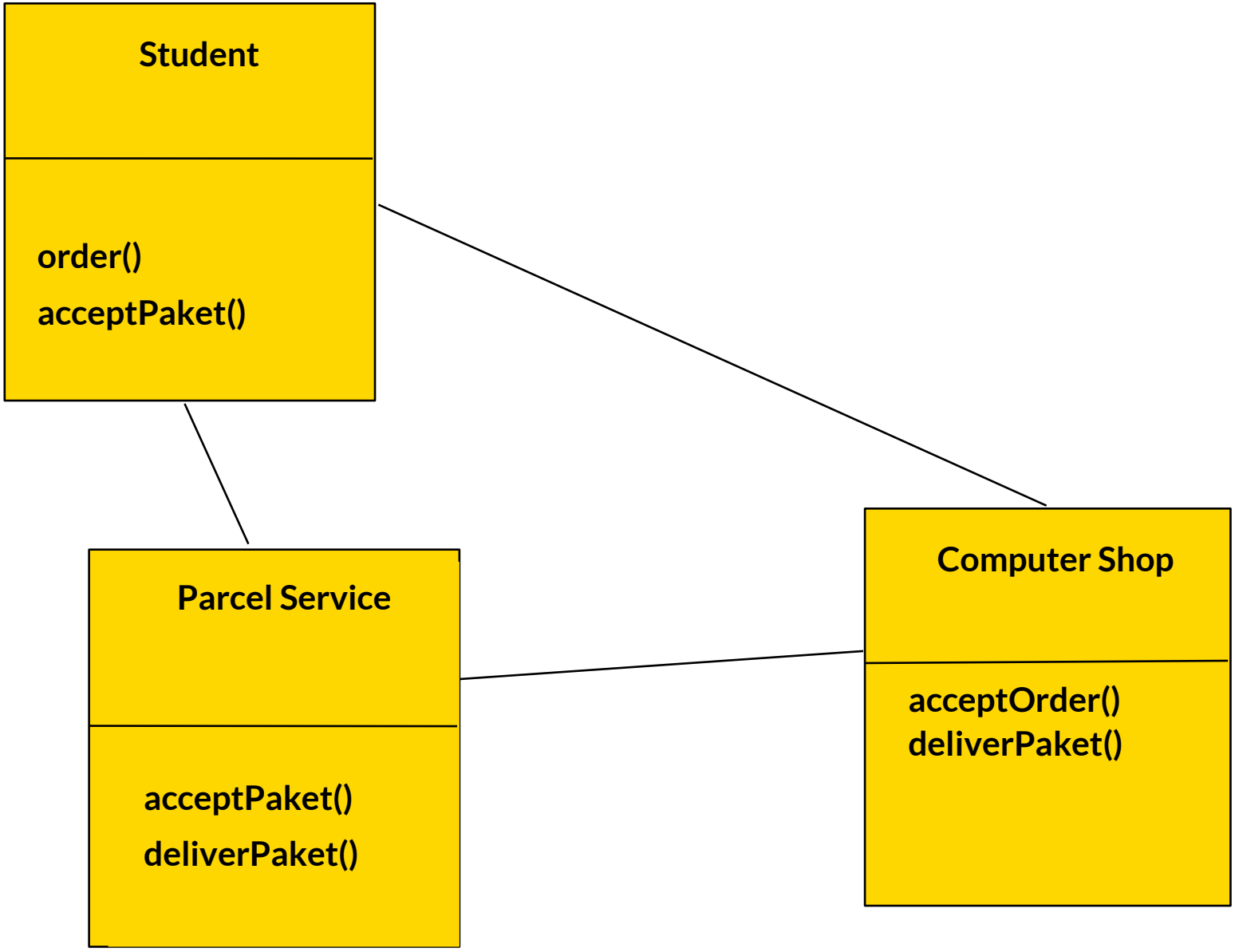
Beispiel: Student bestellt Computers

Student	*Cooperating with*
Responsible for	
order Accept paket	Computer Shop Parcel Service

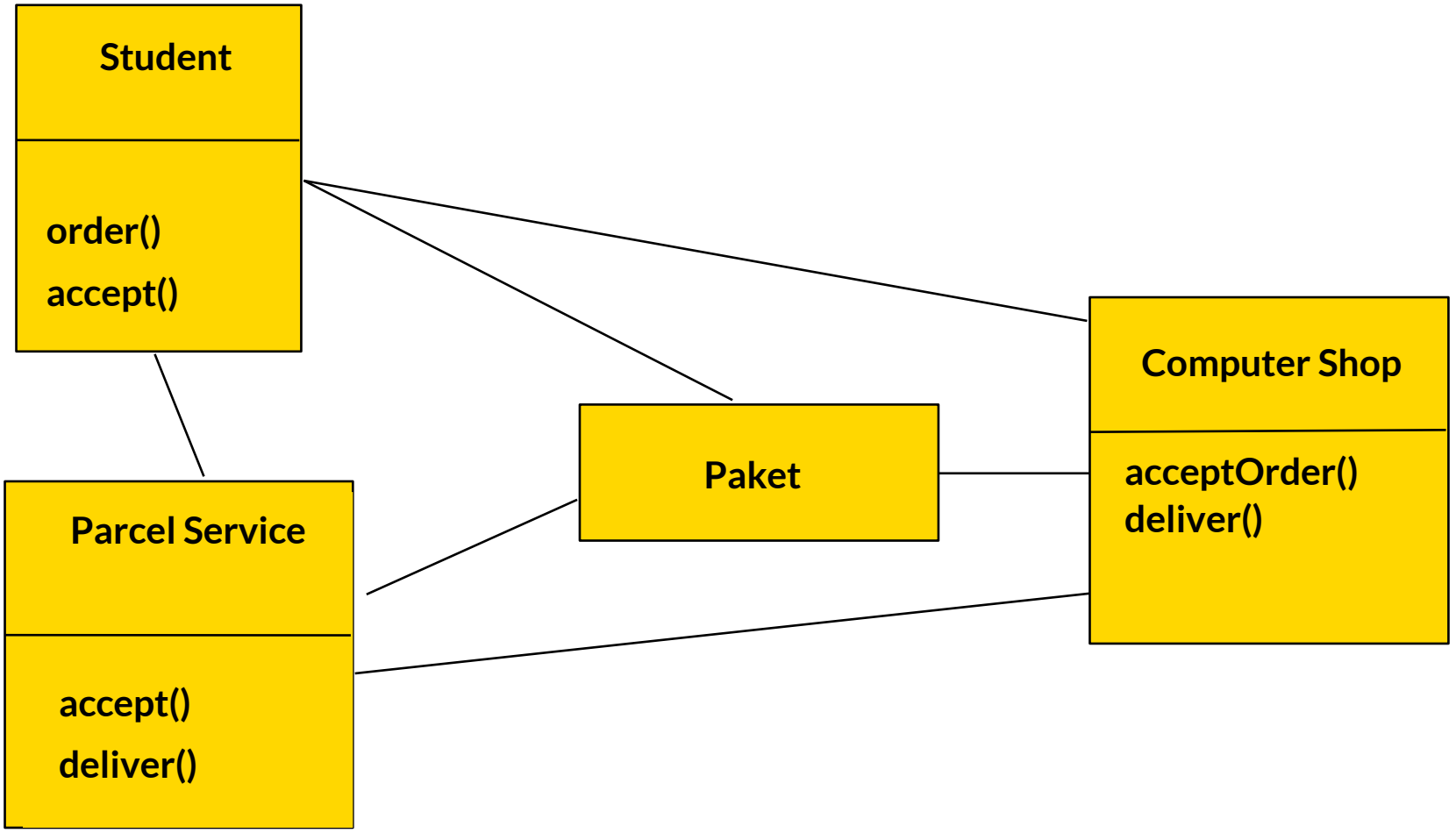
Parcel Service	*Cooperating with*
Responsible for	
Accept paket Deliver paket	Computer Shop Student

Computer Shop	*Cooperating with*
Responsible for	
Accept Order Deliver paket	Student Parcel Service

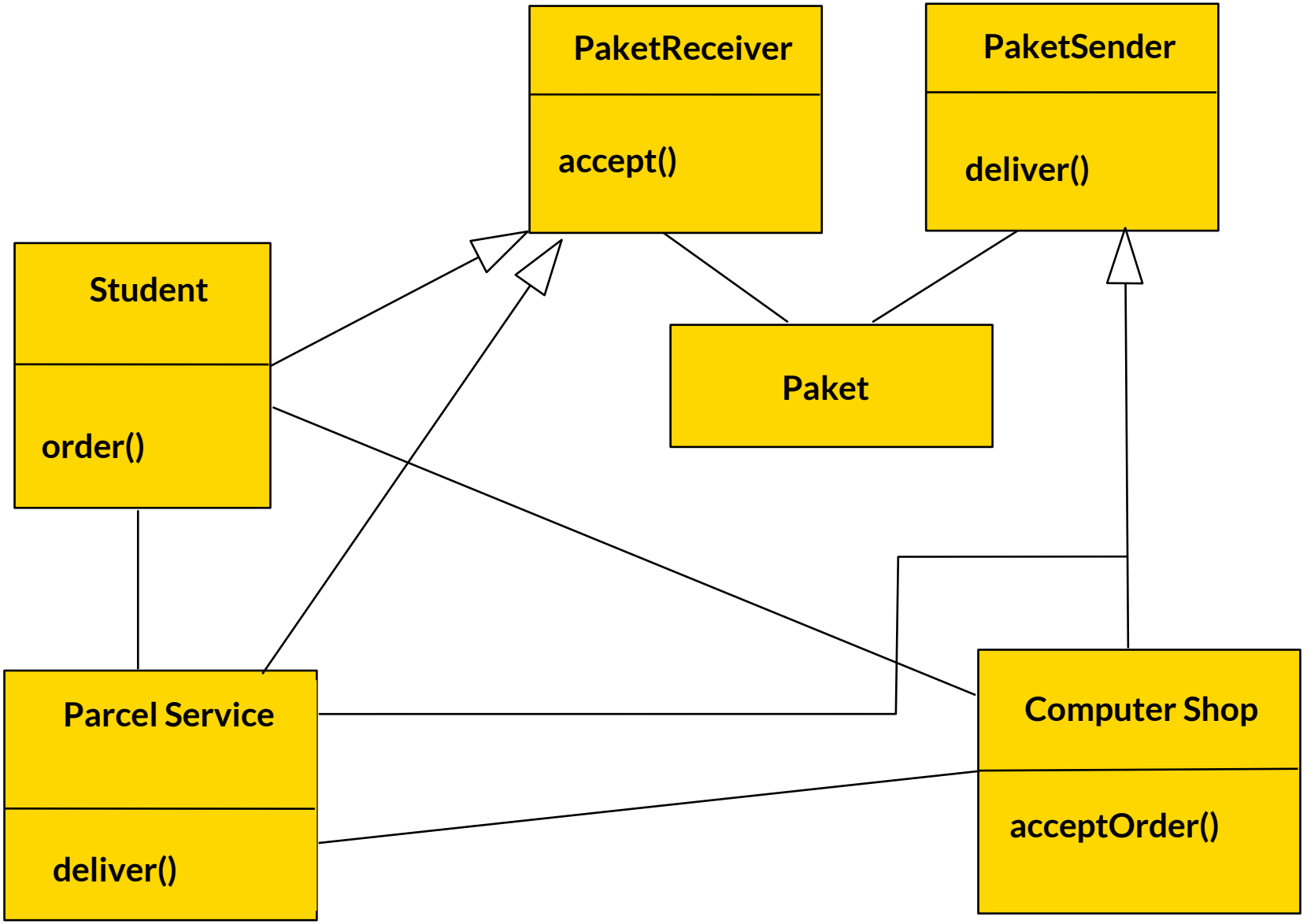
Ein erstes Klassendiagramm



1. Verfeinerung: Pakete als Objekte



2. Verfeinerung: Ausfaktorisieren von Gemeinsamkeiten



Was haben wir gelernt?

- CRC-Karten dienen als Mittel, mit Gedankensturm (brainstorming) die Klassen und ihre Zuständigkeiten herauszufinden.
- Verb-Substantiv-Analyse von Kundentexten hilft, die Objekte und ihre Klassen zu finden
- Achte auf den korrekten Kontext von Klassen
- Gruppenspiele dienen zum iterativen, reflektiven Finden von Klassen und Zuständigkeiten
  - Beachte die verschiedenen Rollen der Gruppenmitglieder

The End

- Warum muss man sich merken, aus welcher User Story (textuellem Requirement) eine CRC-Karte entstanden ist?
- Wie wandelt man CRC-Karten in ein Klassendiagramm-Snippet?
- Warum können CRC-Karten immer nur ein Snippet eines Klassendiagramms ausdrücken?

12. Erste Schritte in der Objektorientierte Analyse mit CRC-Karten

Prof. Dr. rer. nat. Uwe Aßmann
Lehrstuhl Softwaretechnologie (ST)
Institut für Software- und Multimediatechnik (SMT)
Fakultät für Informatik
TU Dresden
Version 22-0.1, 30.04.22

1) CRC-Analyse
2) CRC mit Gruppen
3) CRRC-Analyse

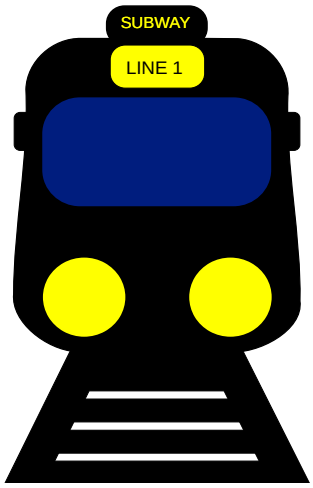
Bitte selbständig studieren!
Begleitmaterial zur Analyse
Wird in Kap 31 noch mal aufgenommen

Literatur

- Obligatorische Literatur
  - Zuser Kap 9
  - Beck, Kent; Cunningham, Ward (October 1989), "A laboratory for teaching object oriented thinking", ACM SIGPLAN Notices (New York, NY, USA: ACM) 24 (10): 1–6, http://c2.com/doc/oopsla89/paper.html
- Weiterführende Literatur
  - HotDraw CRC cards http://c2.com/doc/crc/draw.html
  - Scott Ambler. The Object Primer. Cambridge University Press. Gutes Kapitel über CRC
  - [PP] Andrew Hunt, David Thomas. The pragmatic programmer. Addison-Wesley. Deutsch: Der Pragmatische Programmierer. Hanser-Verlag. Leseprobe
    - http://www.beck-shop.de/fachbuch/leseprobe/9783446223097_Excerpt_004.pdf

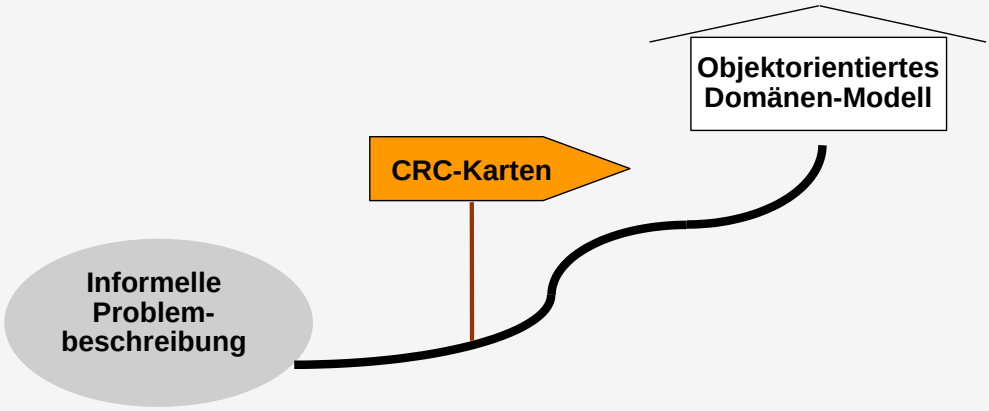
Q 0.1 Java Herunterladen

- Das Java Development Kit (JDK)
- https://adoptopenjdk.net/, brew tap adoptopenjdk/openjdk
- http://openjdk.java.net/, brew info openjdk



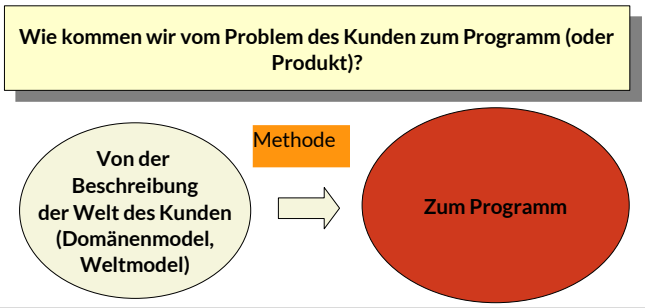
If you have not yet downloaded Java, and started the compiler and the VM, you are already rather late and in danger to miss the train.

12.1 Analysemethoden: Analyse mit CRC-Karten



Die Rolle von Methoden in der Softwareentwicklung

- Um ein System zu entwickeln, sollte man sich an eine ***Softwareentwicklungs-Methode*** halten, die einem durch alle Schritte leitet
  - Eine Methode beruht auf einer oder mehreren zentralen Fragen, die immer wieder gestellt werden
  - Analysemethode – Entwurfsmethode - Implementierungsmethode



Sie werden im Laufe Ihres Lebens verschiedene Softwareentwicklungsmethoden kennenlernen. Neben der Objektorientierten Methode gibt es

- Funktionale Entwicklung
- Zustandsbasierte Entwicklung
- Ereignis-Bedingungs-Aktionsbasierte Entwicklung
- Datengetriebene Entwicklung

Siehe Softwaretechnologie II (5. Semester)

Objektorientierte Entwurfsmethodik

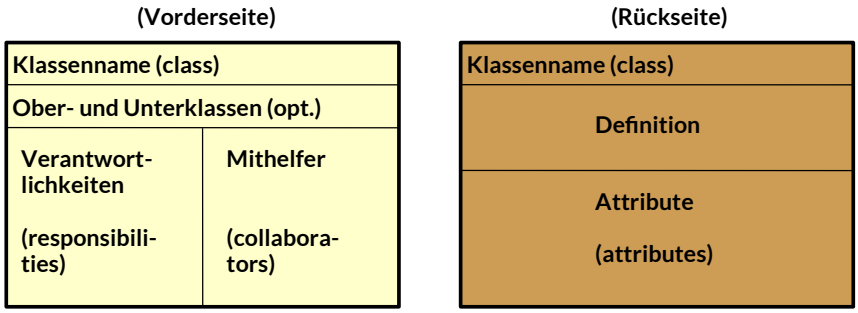
- Die die Entwicklung treibenden Fragen sind:

Welche Objekte (Klassen) enthält ein System?
Welche Verantwortlichkeiten haben sie?

- Vom objektorientierten Entwurf existieren einige Spielarten, die zusätzliche Hilfsmittel einsetzen.
- CRC-Karten
- Strukturgetriebener Klassenentwurf (z.B. nach Balzert, später)

Analyse mit CRC-Karten

- ***CRC*** = **C**lass – **R**esponsibility – **C**ollaborator
  (Klasse – Verantwortlichkeit – Mithelfer)
- [Beck, Cunningham, Wilkerson, Wirfs-Brock (ca. 1989-1995)]
- Technik zur Gruppenarbeit (Rollenspiele)
- Wichtigstes Hilfsmittel: Zu beschriftende Karteikarten



Weshalb gerade Karteikarten?

- Kostengünstig, einfach zu handhaben
- Bewährtes Werkzeug beim Organisieren unstrukturierter Information
- Physischer Umgang mit Karten besser für Gruppenatmosphäre als z.B. Software-Werkzeug
- Größe der Karteikarten begrenzt Komplexität der Klassen
- "Spielen" mit den Karten (Heben der Rollenkarte) erleichtert Identifikation

(nach: Wilkinson 95)

Vorder- und Rückseite sind "zwei Seiten einer Medaille":

Voderseite: Verhaltensorientierte Sicht, Operationen, dynamische Aspekte

Rückseite: Datenorientierte Sicht, Attribute, statische Aspekte

Auf das Konzept von Ober- und Unterklassen bei CRC-Karten wird im folgenden nicht weiter eingegangen. Es ist durchaus auch möglich, CRC-Karten ohne diese Einträge anzuwenden.

Class Responsibility Cards (CRC)

Welches Objekt ist für welche Aufgaben zuständig?

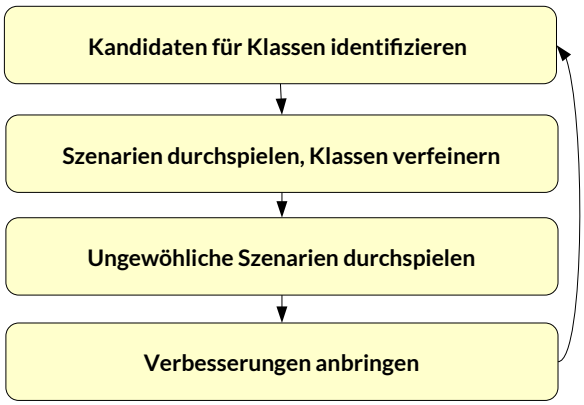
- Zuständigkeit für Aufgabe (Dienst):
  - Aktion
  - Auskunft (query)
  - Prüfung (check)
- Kooperation mit Partner
  - Wenn Klasse alleine zur Bewältigung der Aufgaben nicht fähig
  - “Mit wem muss ich kooperieren, um einen Dienst zu erhalten?”

(Vorderseite)

Klassenname (class)	
Ober- und Unterklassen	
Verantwort-lichkeiten **(responsibili-ties)**	**Mithelfer** **(collabora-tors)**

CRC-Karten-Methode: Vorgehensweise als Einzelner

- Voraussetzung: informelle Anforderungsbeschreibung
  (ideal: ausführliche Anforderungsspezifikation)



Gesetz 51 [PP]: Nicht Anforderungen sammeln, sondern nach ihnen schürfen

Szenarien

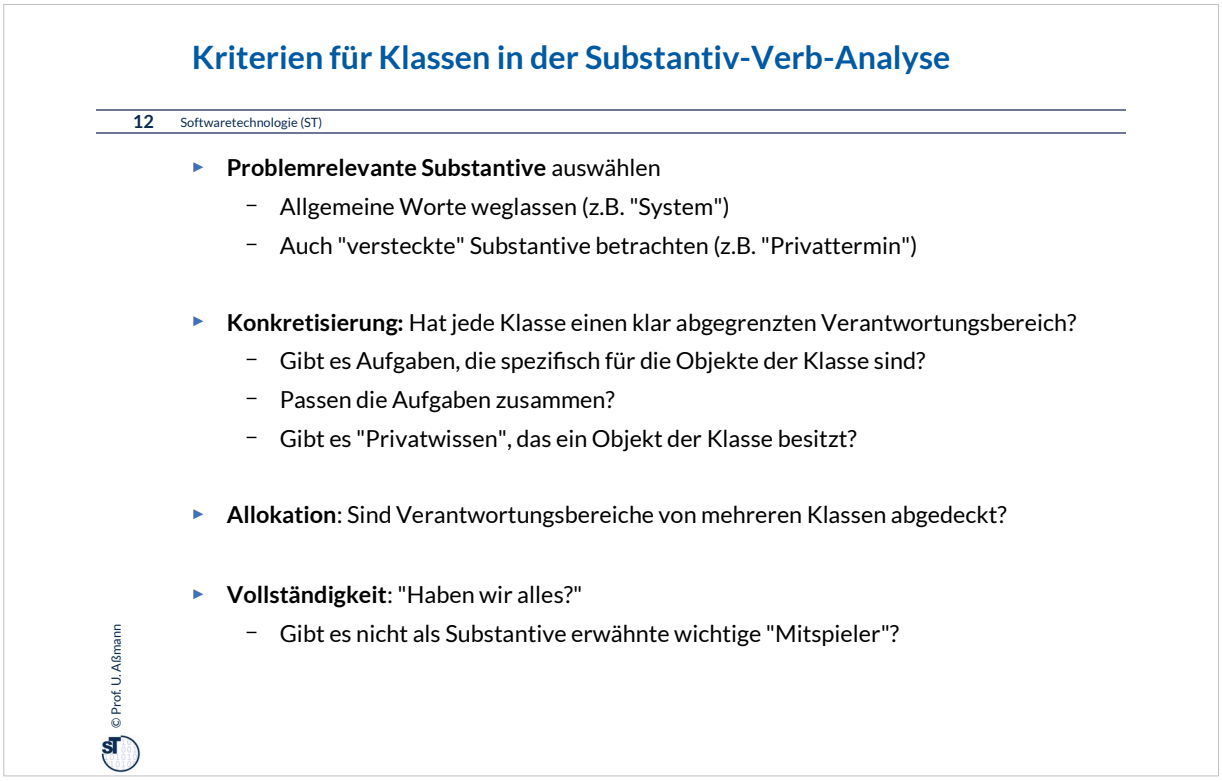
- Ein **Szenario** ist ein typischer Ablauf von Aktionen zur Erfüllung des geplanten Systemzwecks.
  - z.B. notwendige Schritte zur Einrichtung einer neuen Teambesprechung:
    - Festlegung des Titels
    - Festlegung der Teilnehmer
    - Festlegung eines ersten Terminvorschlags und Abstimmung
    - Festlegung eines Besprechungsraums
  - Viele Szenarien zur Beschreibung eines Teilaspekts des Systemzwecks.
- Szenarien werden interaktiv "durchgespielt". Man stellt sich die Entwicklungsfragen der CRC-Methode:
  - Wer ist zuständig (Verantwortungsbereich)?
  - Welche Aufgaben sind dazu zu erfüllen? (auf Karte eintragen)
  - Welches Privatwissen ist dazu nötig? (auf Karte eintragen)

Identifikation von Klassen-Kandidaten: Substantiv-Verb-Analyse

- Analysiere textuelle Anforderungen:
  - Finde *Objekte* aus Hauptwörtern
  - Finde *Kooperationen* aus Subjekt-Objekt-Relationen, Genitiven, Nebensätzen
  - Finde *Aktivitäten* aus Verben und substantivierten Verben
  - Finde *Kontextklassen* durch Orte und adverbiale Bestimmungen

Finde Aufgaben aus Verben, Klassen aus Hauptwörtern

- Beispiel
- "When the student orders a computer, he has to contact the computer dealer. The computer dealer ships the packet *via the parcel service*."
- "When the driver turns on the lights the battery is discharged. When the engine runs the dynamo recharges the battery..."



Verantwortlichkeiten:

•Eine Verantwortlichkeit enthält fast immer ein Verb (Zeitwort).

•'... wissen' kann auch eine Verantwortlichkeit sein.

Mithelfer:

•Nur eintragen, wenn Kommunikation mit anderen Objekten notwendig.

•Eine Verantwortlichkeit kann mehrere Mithelfer benötigen.

•Die Rückgabe einer Antwort gehört zu einem normalen Kommunikationsvorgang - nicht als Verantwortlichkeit nennen.

Karten-Rückseiten:

•Definitionen am besten vor dem Spiel ausfüllen, später überprüfen.

•Attribute können während des Spiels oder später ausgefüllt werden.

Klassen finden: Beispiel Terminverwaltung

Problembeschreibung (user story):

Es ist ein Terminverwaltungssystem für Arbeitsgruppen zu entwickeln.

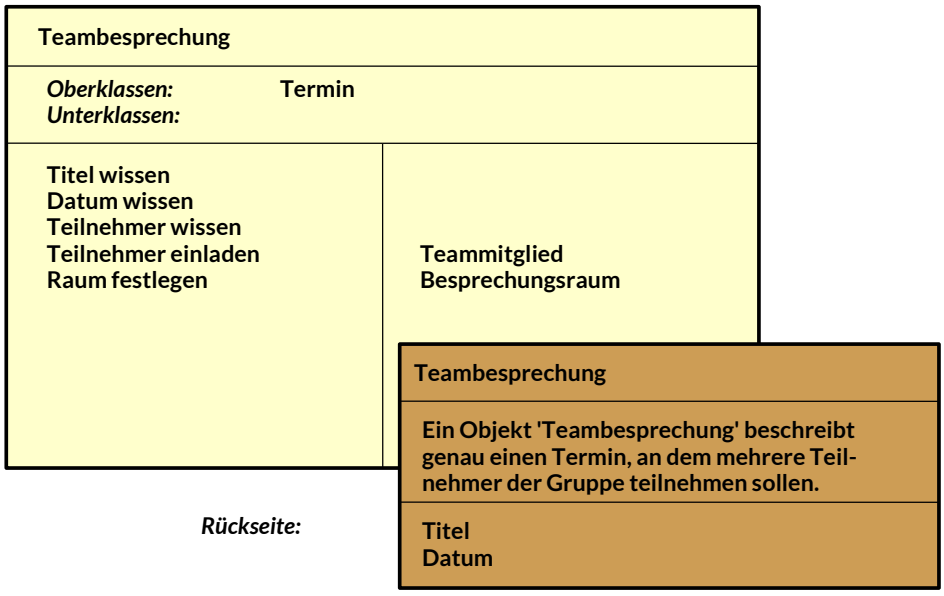
Das System soll alle geplanten Teambesprechungen (z.B. Projektbesprechungen) speichern und die Reservierung von Besprechungsräumen unterstützen.

Das System soll automatisch Kollisionen mit bereits bekannten Terminen vermeiden. Deshalb soll auch die Eintragung privater Termine möglich sein.

Erste Hinweise für Karten geben die problemrelevanten Substantive im Text:

- Terminverwaltung, Arbeitsgruppe, Teambesprechung,
- Projektbesprechung, Reservierung, Besprechungsraum,
- Termin (mehrfach), Eintragung, Kollision, privater Termin

Beispiel einer CRC-Karte



Handelt es sich um einen klar abgegrenzten Verantwortungsbereich?

- Streichen: Terminverwaltung, Eintragung

Sind Verantwortungsbereiche mehrfach genannt?

- Reservierung kann hier entfallen (nicht, wenn andere
- Ressourcen, z.B. Geräte, reserviert werden müssen!)

"Haben wir alles?"

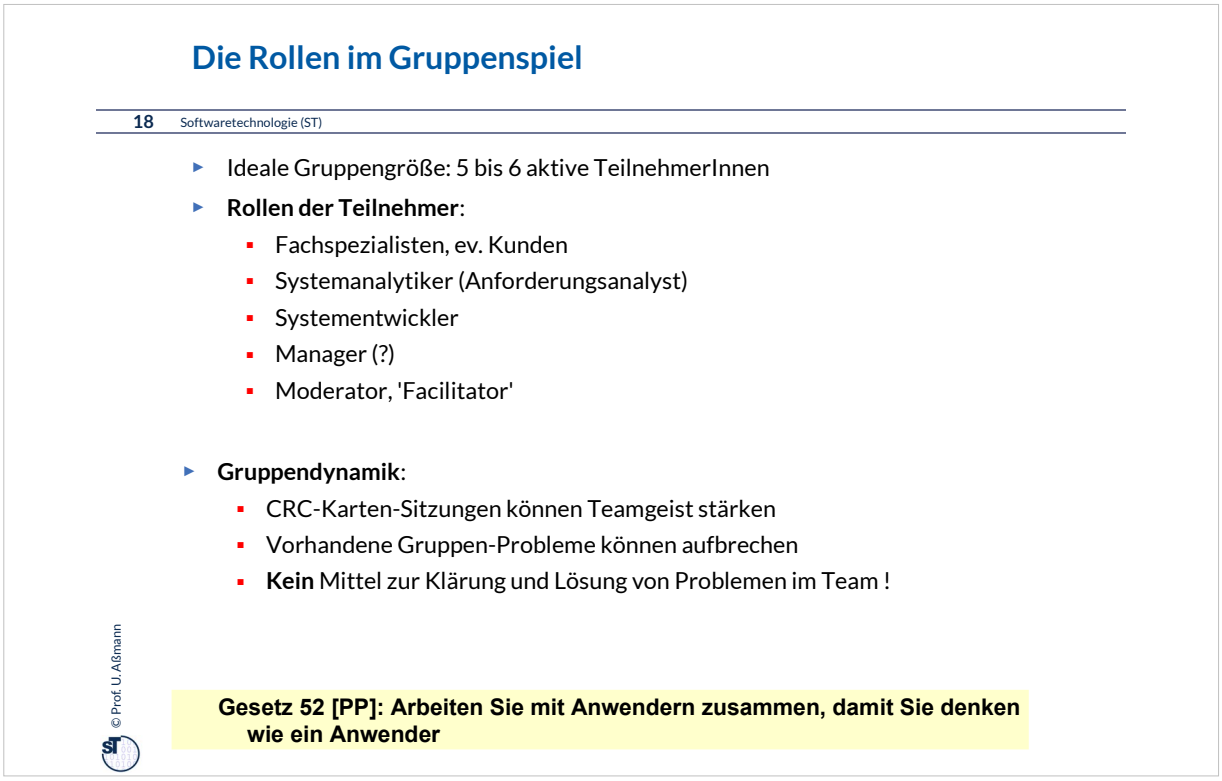
- Projektbesprechung -> Projekt als Klasse
- privater Termin -> MitarbeiterIn

12.2. CRC in Gruppen

CRC-Karten-Methode: Vorgehensweise in Gruppe

- Das Kernstück der Methode sind intensive Gruppensitzungen.
- **Voraussetzung**: informelle, textuelle Anforderungsbeschreibung (ideal: ausführliche Anforderungsspezifikation)
  - Kandidaten für Klassen (Karten) identifizieren
  - Karten auf einem Tisch oder Whiteboard plazieren.
    - Kollaborierende Karten nah zueinander plazieren, andere voneinander entfernen. Anordnung ständig ändern, je mehr Zusammenarbeiten zustandekommen
    - “Heisse” Karten in die Mitte des Tisches
- **Durchführung**
  - Typische Szenarien identifizieren und durchspielen (dabei: Karten schrittweise ausfüllen)
  - Iteration: Verbesserungen, mehrfache Wiederholung
  - Ungewöhnliche Szenarien durchspielen

Gesetz 17 [PP]: Programmieren Sie nahe am Problem (an der Sprache der Anwender)



Auszug aus einem möglichen Gruppenspiel:

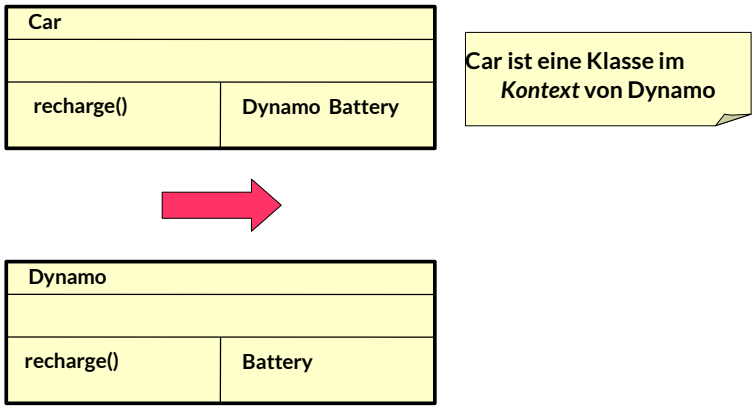
Teilnehmer haben Karten unter sich aufgeteilt.

Wer eine Karte hebt, 'ist dran'.

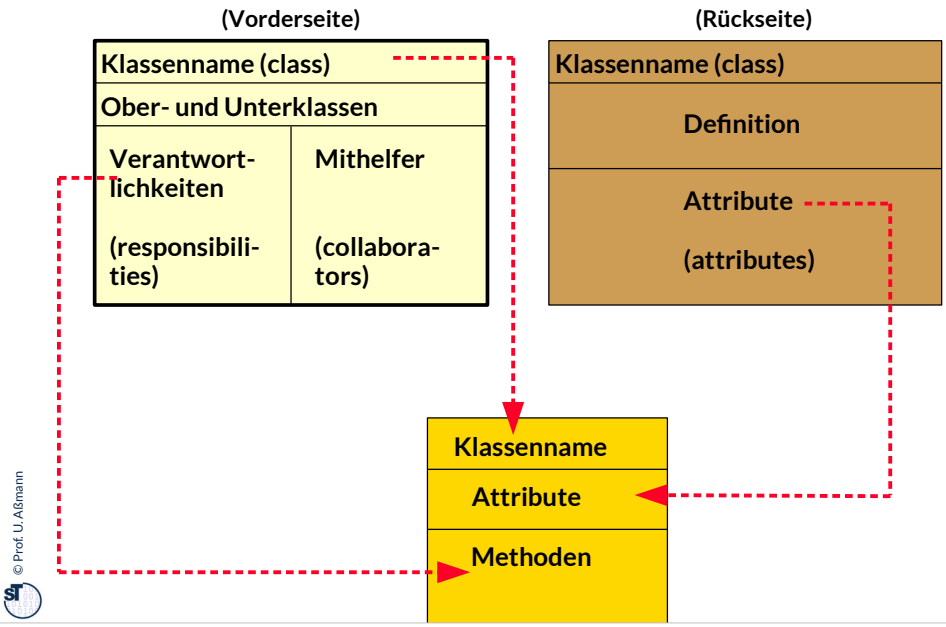
- Moderator: *Es wird eine neue Teambesprechung vereinbart, mit dem Titel 'Erfahrungen mit CASE-Einsatz'.*
- Halter der Karte 'Teambesprechung': *Das ist meine Zuständigkeit. Ich merke mir den Titel.*
- Moderator: *Nun kommt die Teilnehmerliste und ein Terminvorschlag.*
- Teambesprechung: *OK, ich merke mir das Datum. Ich frage nun jeden einzelnen Teilnehmer, ob er/sie Zeit hat. Wer ist zuständig für die Teilnehmer?*
- Teilnehmer Franz Müller: *Ich bin auf der Teilnehmerliste und kann auch zusagen.* (usw.)
- Teambesprechung: *Jetzt suche ich einen geeigneten Raum. Wer hat den Überblick über alle Besprechungsräume?*

Vorsicht: Klassen des Kontexts

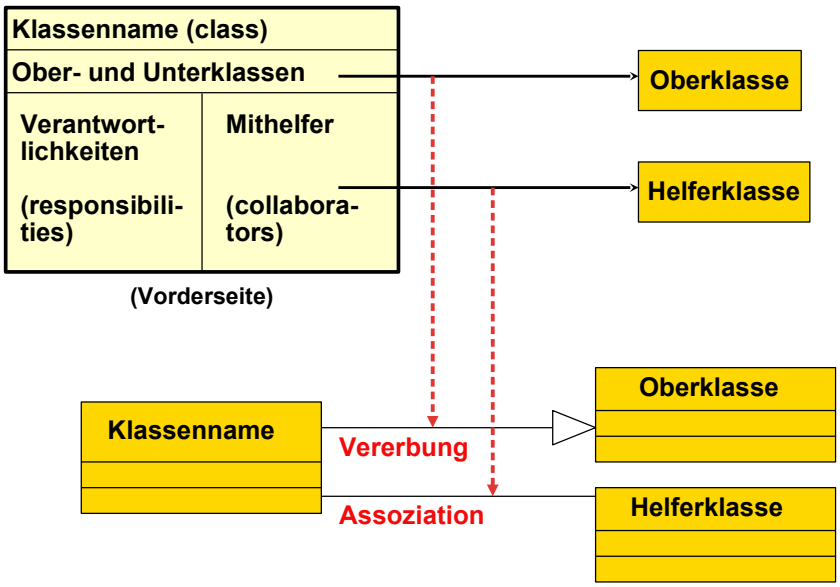
- Oft werden Verantwortlichkeiten fälschlicherweise einer Klasse zugeordnet, die eigentlich eine Klasse im *Kontext* ist
- Achte auf Ortsbestimmungen und Urheberschaften:
  - "the car recharges the battery *via the dynamo*"
- Achte auf Passivsätze



Von CRC-Karten zum UML-Klassenmodell (1)



Von CRC-Karten zum UML-Klassenmodell (2)



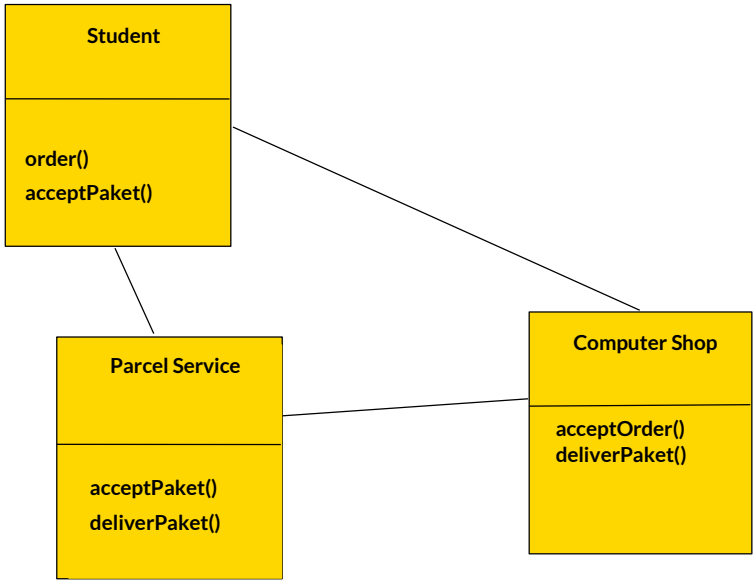
Beispiel: Student bestellt Computers

Student	*Cooperating with*
Responsible for	
order Accept paket	Computer Shop Parcel Service

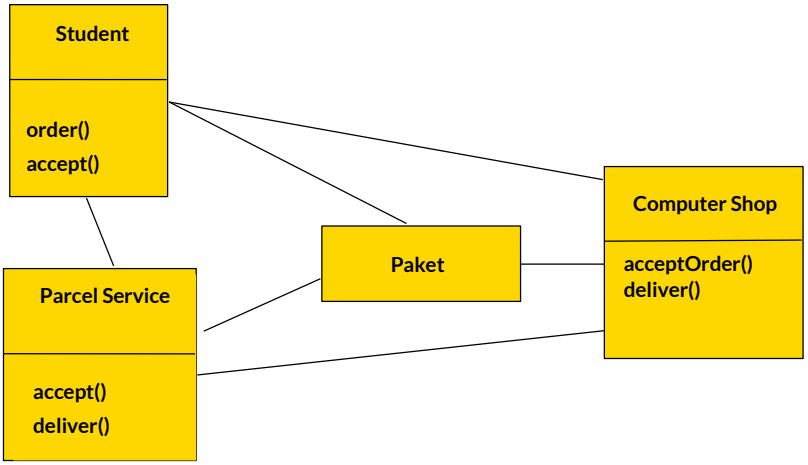
Parcel Service	*Cooperating with*
Responsible for	
Accept paket Deliver paket	Computer Shop Student

Computer Shop	*Cooperating with*
Responsible for	
Accept Order Deliver paket	Student Parcel Service

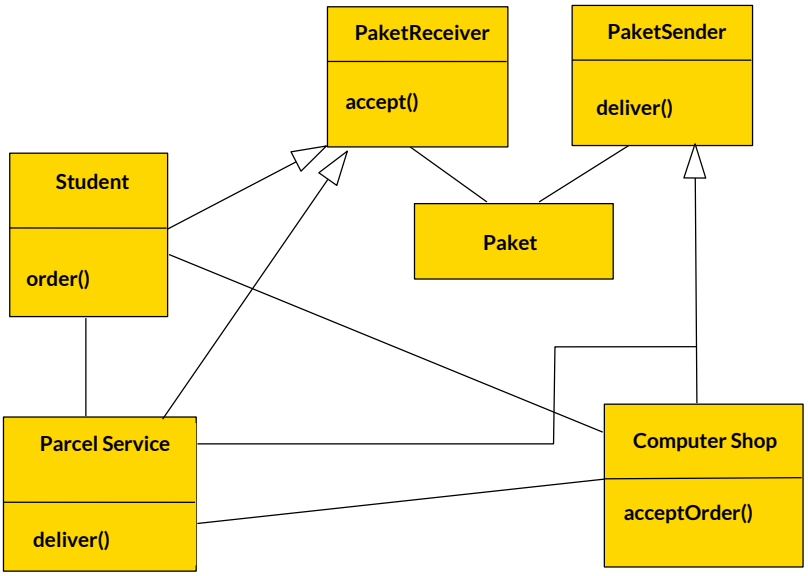
Ein erstes Klassendiagramm



1. Verfeinerung: Pakete als Objekte



2. Verfeinerung: Ausfaktorisieren von Gemeinsamkeiten



Was haben wir gelernt?

- CRC-Karten dienen als Mittel, mit Gedankensturm (brainstorming) die Klassen und ihre Zuständigkeiten herauszufinden.
- Verb-Substantiv-Analyse von Kundentexten hilft, die Objekte und ihre Klassen zu finden
- Achte auf den korrekten Kontext von Klassen
- Gruppenspiele dienen zum iterativen, reflektiven Finden von Klassen und Zuständigkeiten
  - Beachte die verschiedenen Rollen der Gruppenmitglieder

The End

- Warum muss man sich merken, aus welcher User Story (textuellem Requirement) eine CRC-Karte entstanden ist?
- Wie wandelt man CRC-Karten in ein Klassendiagramm-Snippet?
- Warum können CRC-Karten immer nur ein Snippet eines Klassendiagramms ausdrücken?