ソフトウェア工学 第5回
― オブジェクト指向分析 ―

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

講義内容

- オブジェクト指向
  - オブジェクト、クラス
  - 関連、継承、集約
- オブジェクト指向分析
  - ユースケース図
  - クラス図
  - アクティビティ図
  - シーケンス図
  - 状態機械図

オブジェクト指向 (object-oriented)

- 現実世界のモデルをソフトウェアで直接的に表現する一つの方法
  - オブジェクトによるモデリング
  - 人間の認知方法にできるだけ近づけることを目指した技法

    → 人間にとって理解しやすい
  - 開発において、一貫してオブジェクトを中心に考える

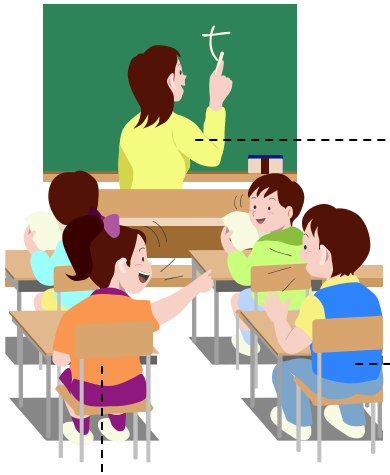
    → 分析、設計、実装が比較的容易に移行可能

オブジェクト

- object　＝(辞書的には、)物
  - 人間が認知できる具体的あるいは抽象的な「物」
  - 実世界の物体や役割、概念を抽象化した「物」

- オブジェクトの持つ特性
  - 状態 (state)
    - オブジェクトの現在の性質
    - プログラミング言語により属性(attribute)、プロパティ(property)等
  - 振る舞い (behavior)
    - オブジェクトが実行できる動作
    - プログラミング言語により操作(operation)、メソッド(method)等
  - 識別性(しきべつ) (identity)
    - 個々のオブジェクトを区別する手段

属性と操作



教員

属性
- 教員番号
- 氏名
- 住所
- …

操作
- 教員番号?
- 氏名?
- 話す
- 授業を行う
- …

学生

属性
- 学籍番号
- 氏名
- 住所
- …

操作
- 学籍番号?
- 氏名?
- 立つ
- 座る
- 話す
- 宿題をする
- …

学生

属性
- 学籍番号
- 氏名
- 住所
- …

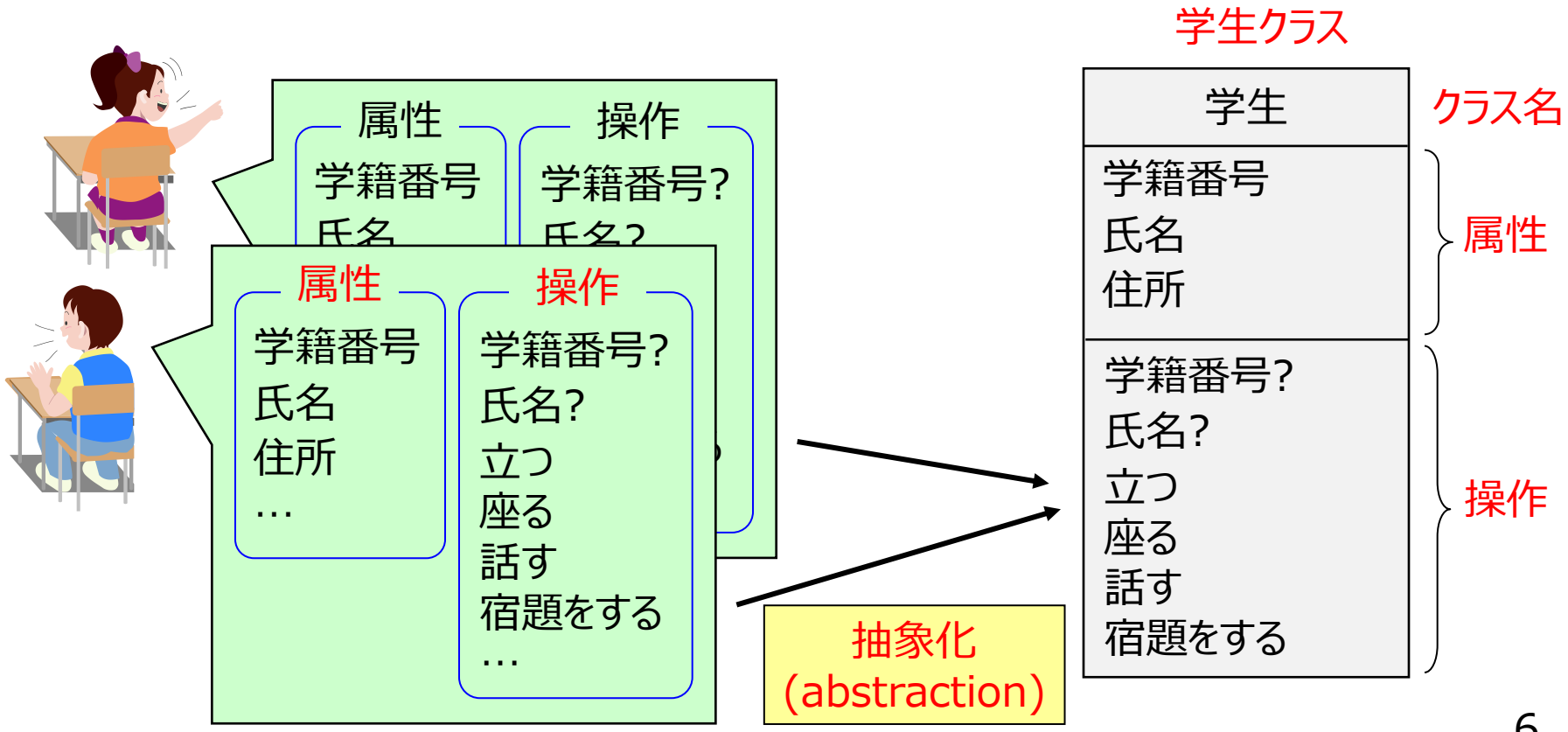
操作
- 学籍番号?
- 氏名?
- 立つ
- 座る
- 話す
- 宿題をする
- …

オブジェクト指向の基本概念

- クラスとインスタンス(class, instance) **実例化**
- カプセル化(encapsulation) **封装**
  - データとそれに対する処理をまとめてモジュール化
  - オブジェクトの状態を外部から隠蔽（情報隠蔽）
- メッセージパッシング(message passing)
  - オブジェクトに処理を依頼する仕組み
- 関連(association)
  - あるオブジェクトが別のオブジェクトを利用することを表す、オブジェクト間の関係
- 継承(inheritance)
  - 既存のクラスに属性や操作を追加して新しいクラスを定義すること
- 集約(aggregation)
  - オブジェクトを構成する部品（部分オブジェクト）を束(たば)ねて扱う仕組み

クラス

- 同じ属性と操作を持つオブジェクトを抽象化したひな形(template) 様本
- オブジェクトの設計図



インスタンス instance

- クラスから生成されたオブジェクト

オブジェクト名の表記方法
オブジェクト名:クラス名

学生クラス

学生
学籍番号 氏名 住所
学籍番号? 氏名? 立つ 座る 話す 宿題をする

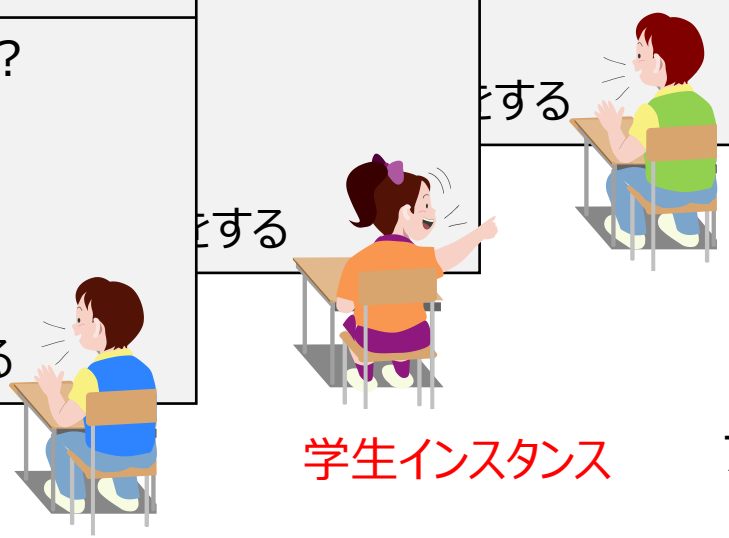
生成
（instantiation）

Ken:学生
学籍番号=“A02” 氏名=“Ken” 住所=“…”
学籍番号? 氏名? 立つ 座る 話す 宿題をする

Miki:学生
学籍番号=“A03” =“Miki” =“…”
番号? をする

Tom:学生
学籍番号=“A04” =“Tom” =“…”
番号? をする

…



学生インスタンス

オブジェクト、クラス、インスタンス

現実

OOモデル

プログラミング

実行環境

Ken

学生

属性

操作

オブジェクトの
認識

学生
学籍番号 氏名 住所
学籍番号? 氏名? 立つ 座る 話す 宿題をする

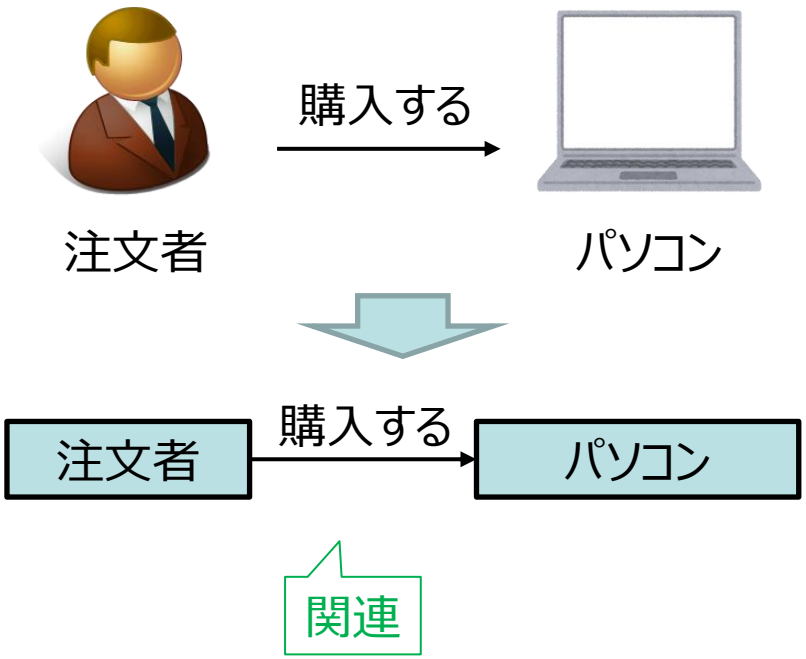
クラスの記述

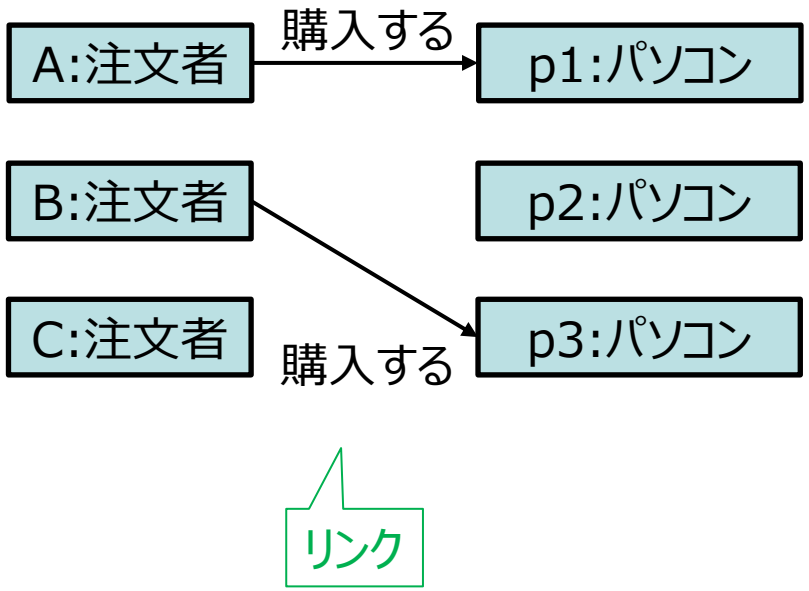
Ken:学生
学籍番号=“A02” 氏名=“Ken” 住所=“…”
学籍番号? 氏名? 立つ 座る 話す 宿題をする

インスタンスの
生成と実行

関連 (association)とリンク(link)

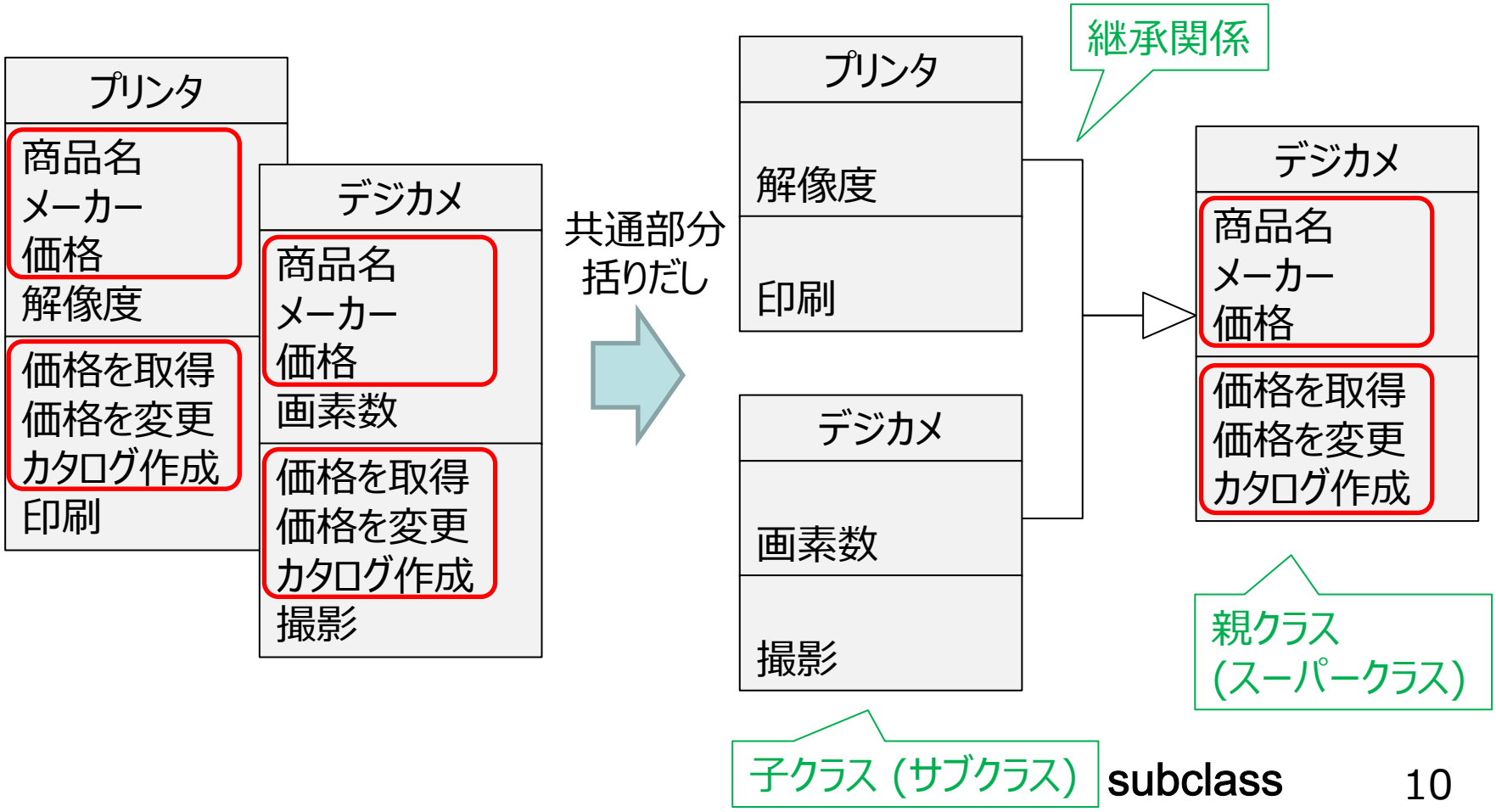
- 関連：クラス間の利用関係
- リンク：オブジェクト間の利用関係





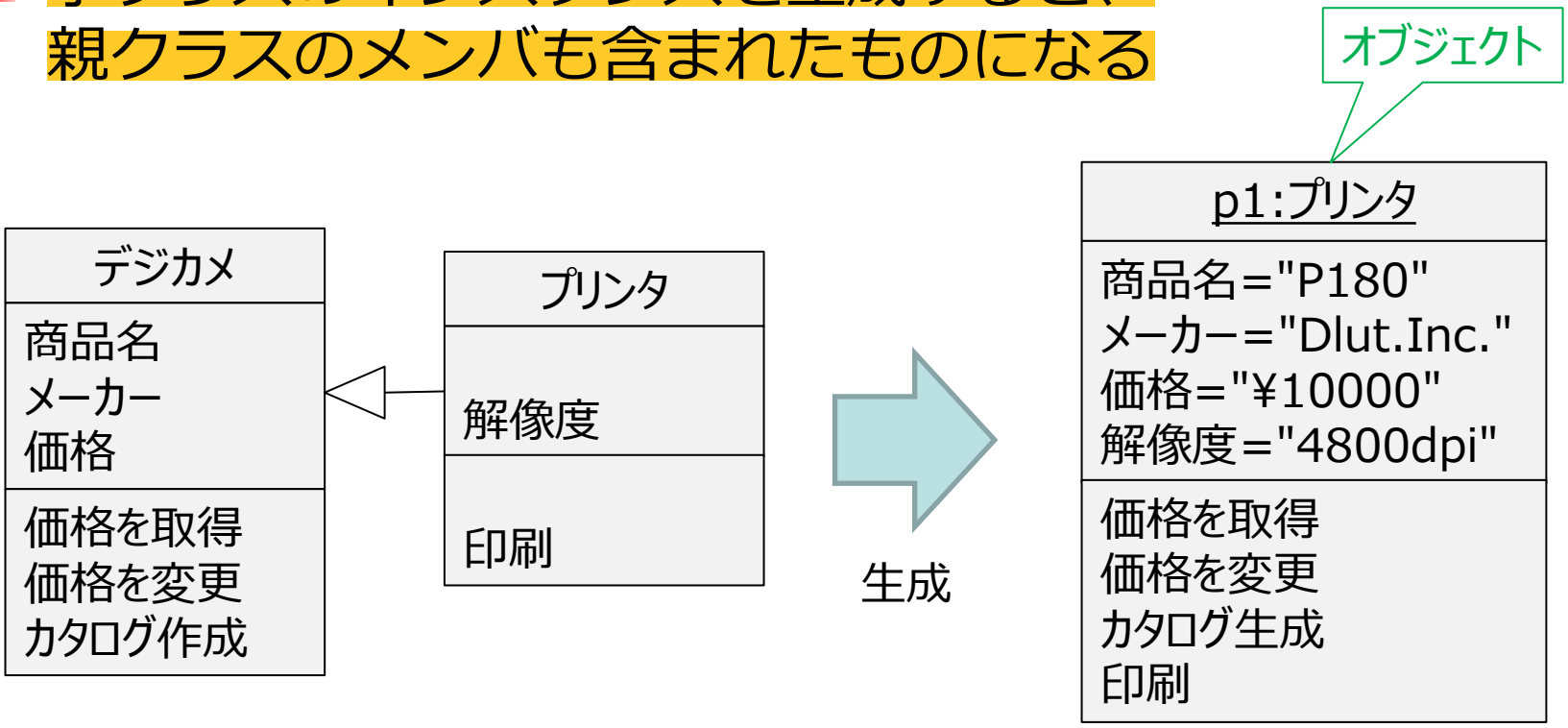
継承 (inheritance)

- 複数のクラスにおける共通な性質を共有させる仕組み



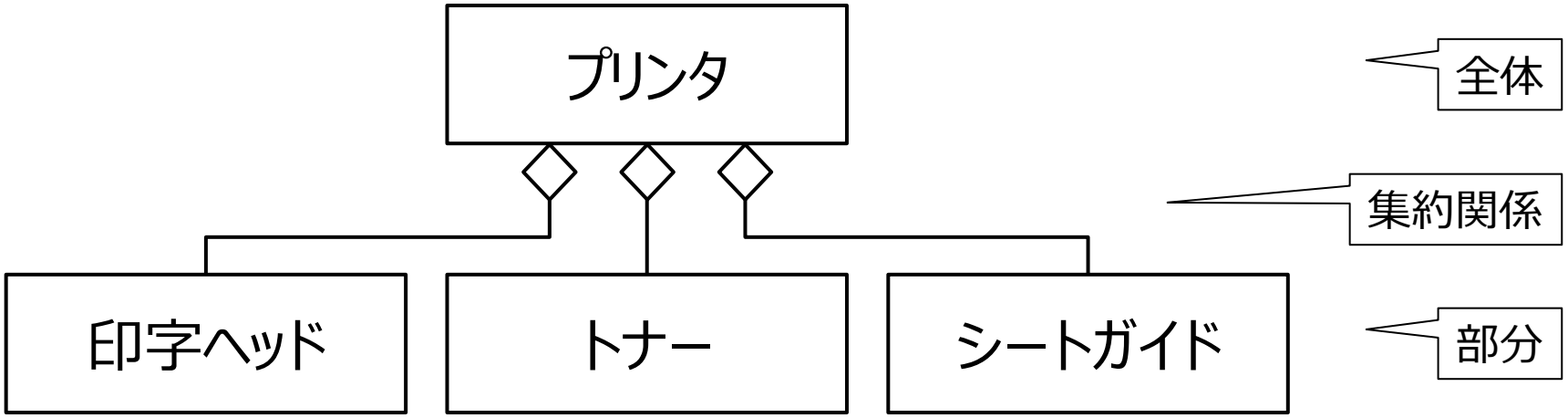
継承とオブジェクトの生成

- 子クラスのインスタンスを生成すると、親クラスのメンバも含まれたものになる



集約 (aggregation)

- 複数のオブジェクトを束ねて扱う仕組み
  - 全体とその構成部品の関係
  - 複合(composition)：強い所有関係を持つ集約
    - 所有物と非所有物の生存期間が一致



聚合是关联关系的一种特例，他体现的是整体与部分、拥有的关系，即has-a的关系，此时整体与部分之间是可分离的，他们可以具有各自的生命周期，部分可以属于多个整体对象，也可以为多个整体对象共享；比如计算机与CPU、公司与员工的关系等

確認問題

- 以下の説明に合う用語を語群から選んで答えよ。
  - オブジェクトの現在の性質を示す。
  - オブジェクトが実行できる動作を示す。
  - 個々のオブジェクトを区別する手段を提供する。
  - 同じ属性と操作を持つオブジェクトを抽象化したひな形。
  - クラスから生成されたオブジェクトのこと。
  - クラス間に利用関係があることを示す。
  - 複数のクラスの共通な性質を共有させる仕組み。
    子クラスは親クラスの性質を引き継ぐ。
  - 複数のオブジェクトを束ねて扱う仕組みで、
    全体とその構成部品の関係を表す。
  - 特に強い所有関係を持つ集約。

クラス、インスタンス、識別性、状態、振る舞い、関連、継承、複合、集約

確認問題

- 以下の説明に合う用語を語群から選んで答えよ。
  - オブジェクトの現在の性質を示す。状態
  - オブジェクトが実行できる動作を示す。振る舞い
  - 個々のオブジェクトを区別する手段を提供する。識別性
  - 同じ属性と操作を持つオブジェクトを抽象化したひな形。クラス
  - クラスから生成されたオブジェクトのこと。インスタンス
  - クラス間に利用関係があることを示す。関連
  - 複数のクラスの共通な性質を共有させる仕組み。
    子クラスは親クラスの性質を引き継ぐ。継承
  - 複数のオブジェクトを束ねて扱う仕組みで、
    全体とその構成部品の関係を表す。集約
  - 特に強い所有関係を持つ集約。複合

クラス、インスタンス、識別性、状態、振る舞い、関連、継承、複合、集約

講義内容

- オブジェクト指向
  - オブジェクト、クラス
  - 関連、継承、集約
- ⇨オブジェクト指向分析
  - ユースケース図
  - クラス図
  - アクティビティ図
  - シーケンス図
  - 状態機械図

オブジェクト指向開発方法論

- オブジェクト指向に基づきソフトウェアを開発するプロセス
  ＋ オブジェクト指向ソフトウェアのモデル化(表記法)

- プロセス
  - オブジェクト指向分析 (OOA: OO analysis)

    オブジェクト指向設計 (OOD: OO design)

    オブジェクト指向プログラミング (OOP: OO programming)
  - 反復型ソフトウェア開発プロセス(iterative, incremental)
    で少しずつ構築する

- モデル化
  - UML (Unified Modeling Language)

Unified Modeling Language (UML)

- モデルを表現する統一的な図式表現法
  - グラフィカルな記法とそのメタモデル(言語の概念)を定義
  - 開発プロセスとは独立

構造	クラス図	クラスの構造(属性や操作)とクラス間の静的な関係
	オブジェクト図	ある時点でのオブジェクトの状態とオブジェクト間の関係
	パッケージ図	パッケージの構成とパッケージ間の依存関係
	複合構成図	実行時のクラスの内部構造
	コンポーネント図	コンポーネントの構造と依存関係
	配置図	システムにおける物理的な配置
振る舞い	ユースケース図	システムの提供する機能と利用者の関係
	アクティビティ図	作業の順序と並行性
	状態機械図	オブジェクトの状態とイベントによる状態遷移
	シーケンス図	オブジェクト間の相互作用の時系列
	コミュニケーション図	オブジェクト間の相互作用のリンク
	タイミング図	オブジェクトの状態遷移のタイミング
	相互作用概要図	シーケンス図とアクティビティ図の概要

オブジェクト指向分析

- システムを構成するオブジェクトの構造や振る舞いを明確にすることで要求を仕様化する技法
  - データ駆動型アプローチ
    - システム内のデータをはじめに認識する
  - 責任駆動型アプローチ
    - システムの振る舞い（責任）をはじめに認識する
    - 近年のUMLを用いたオブジェクト指向分析では、データ駆動型よりも一般的

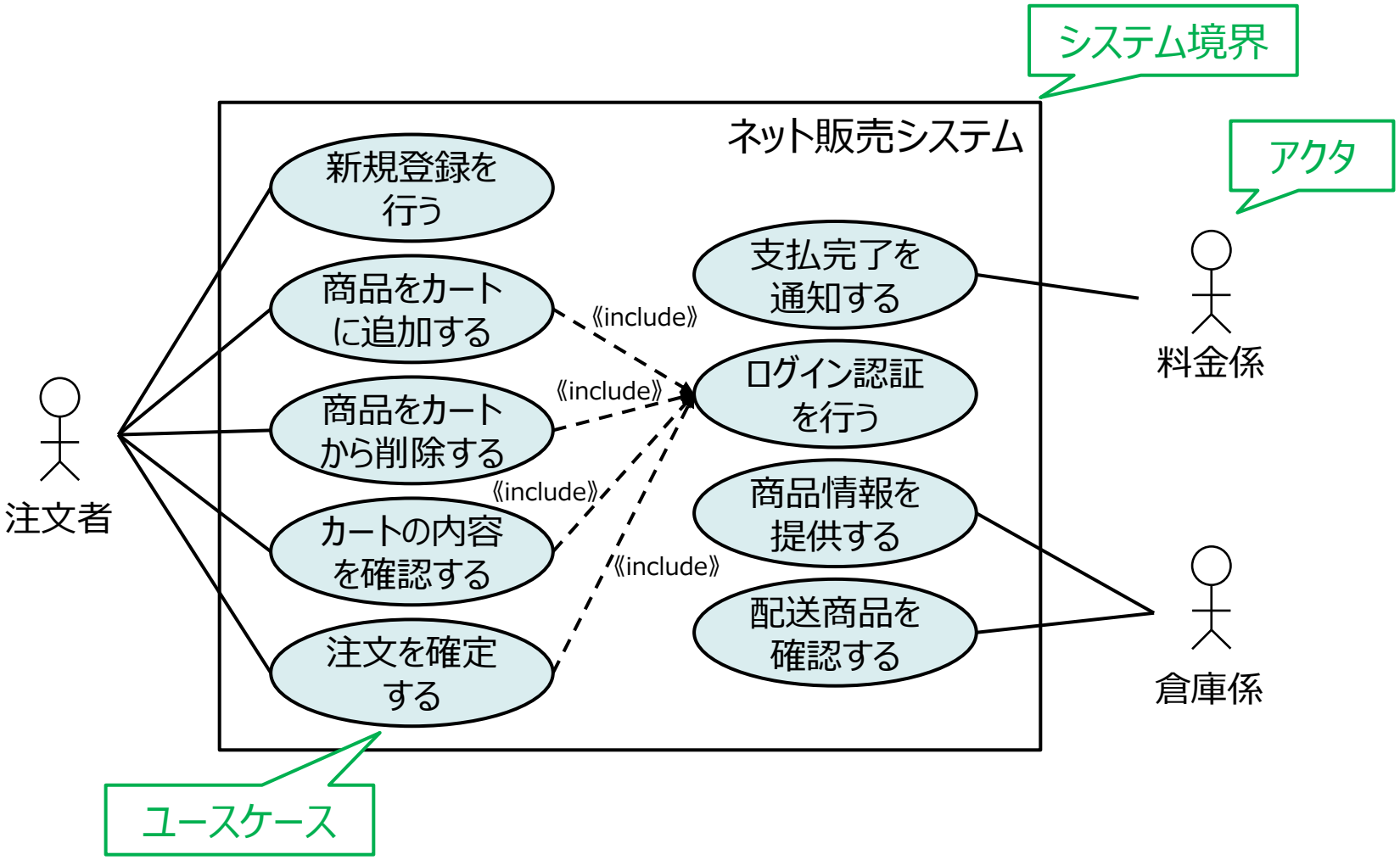
オブジェクト指向分析

- 責任駆動型アプローチにおける手順(例)
  - ユースケース（利用方法）の抽出
    - システムの境界を明確にする
    - アクタ（外部の人間や装置）とユースケースの関係を分析
  - クラスの特定とクラス図の作成
    - システム内の登場人物を洗い出して概念モデルを構築
    - クラスの操作と属性を抽出
  - 振る舞いの記述
    - システム全体の作業の流れの明確化
    - 個々のオブジェクト間の相互作用（メッセージの流れ）の明確化
    - 個々のオブジェクトの状態の明確化
  - モデルの洗練
    - それぞれの図を洗練していく

ユースケース図

- 利用者がどのようにシステムを使用するのかを表す
  - システムの機能ごとに作成
- 主な構成要素
  - アクタ(actor)
    - システムに対して利用者が果たす役割を表現
    - 役割ごとに異なるアクタが存在
    - 外部システムでもよい
  - ユースケース(use case)
    - 外部から見たシステムの振る舞いを表現
  - ユースケース記述
    - 一般的に、シナリオ(scenario)により記述
      - 利用者とシステム間の対話を表す一連の手順
    - 正常系だけでなく異常系(例外処理等)も記述
    - 前提条件や事後条件も記述できる

(例) ユースケース図


システム境界
ネット販売システム
アクタ
新規登録を行う
商品をカートに追加する
商品をカートから削除する
カートの内容を確認する
注文を確定する
支払完了を通知する
ログイン認証を行う
商品情報を提供する
配送商品を確認する
《include》
《include》
《include》
《include》
注文者
料金係
倉庫係
ユースケース

(例) ユースケース記述

名称：　　注文を確定する
開始アクタ：　　注文者
目的：　　カートに存在する商品を購入する
事前条件：　　注文者はログイン認証に成功している

正常処理シナリオ
1. 注文者が、注文確定ボタンを押す。
2. システムは、カート内の商品の合計金額を計算する。
3. システムは、カートの内容と合計金額を表示する。
4. 注文者が、クレジットカード番号を入力し、購入ボタンを押す。
5. システムは、料金係に注文者の支払い状況を確認する。
6. 支払が成功すると、支払完了を通知する。
7. システムは、倉庫係に商品の発送を依頼する。
8. システムは、カートを空にする。
9. システムは、商品の在庫を更新する。

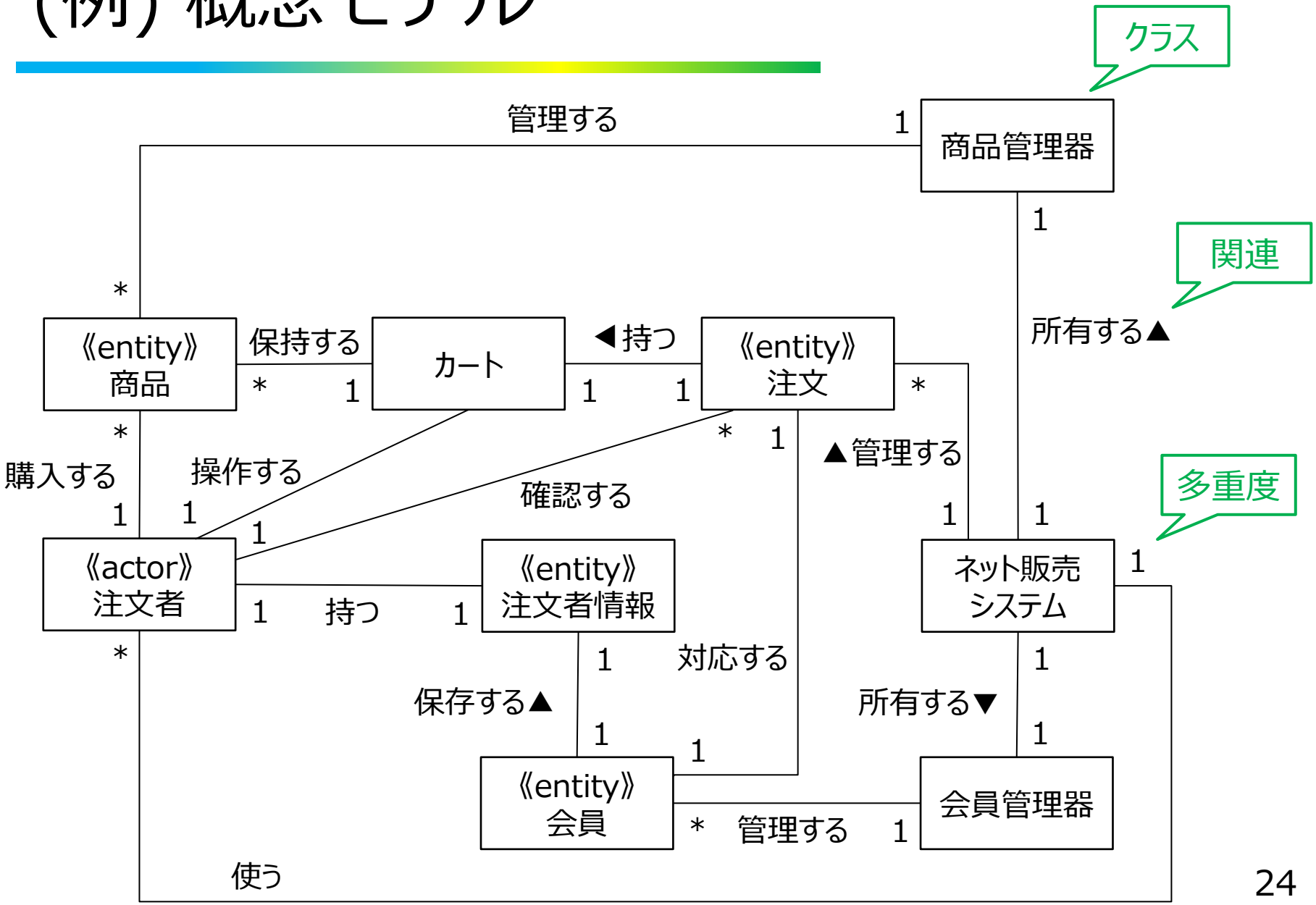
例外処理
5で支払が失敗する。
  a. システムは支払失敗メッセージを表示する。

クラスの特定とクラス図の作成

- システムに登場するクラスを抽出
  - シナリオや要求仕様に出現する名詞が対応することが多い
    - (例) カートは商品を保持する。
- クラス間に存在する関連を抽出
  - シナリオや要求仕様に出現する動詞が対応することが多い
    - (例) カートは商品を保持する。
- 特定したクラスと関連から大まかなクラス図（概念モデル）を作成
- 各クラスの属性と操作を抽出してクラス図を完成
  - 操作が重要

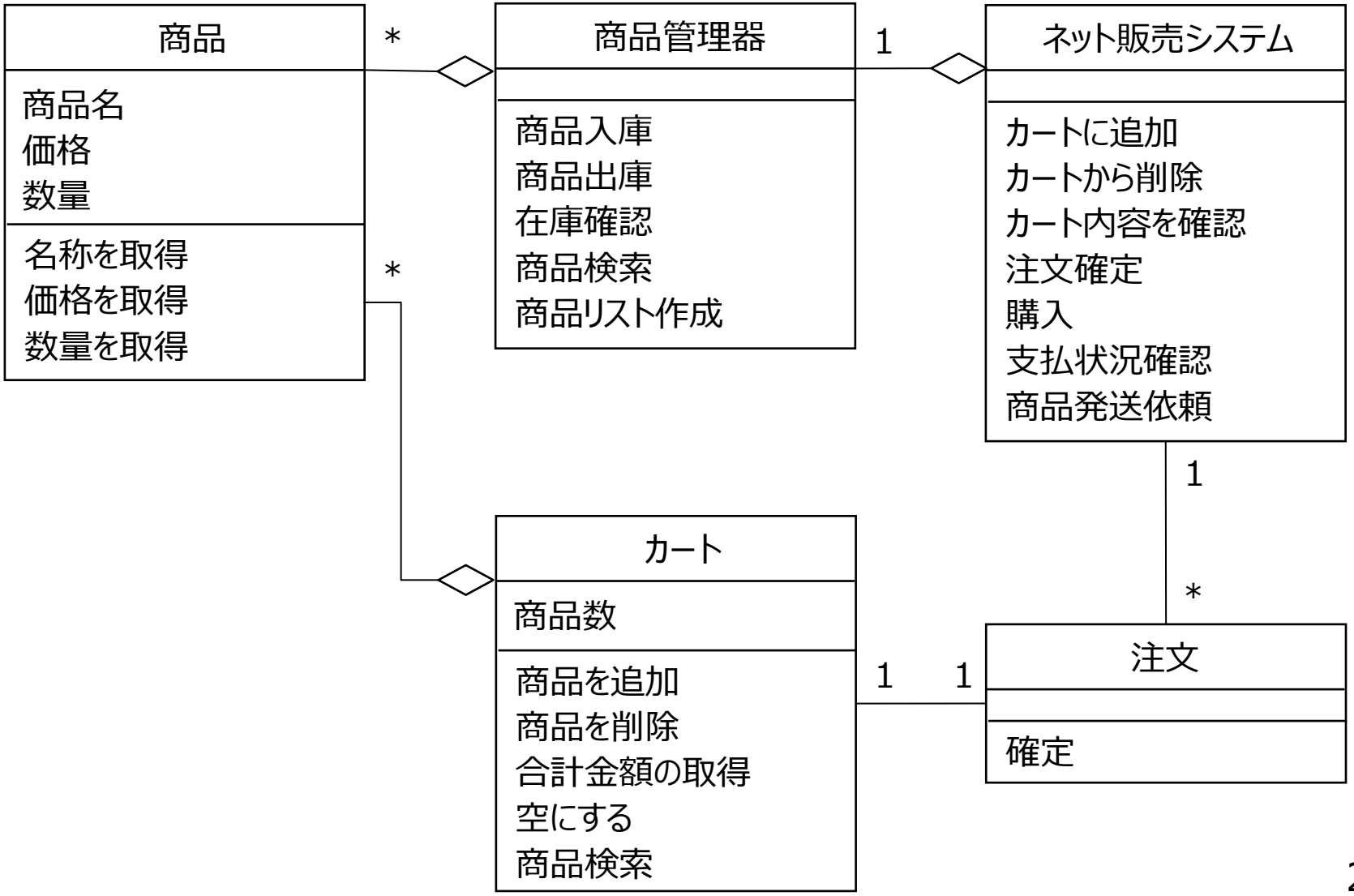
(例) 概念モデル

※前のページのユースケース図から作成したものの一部



(例) クラス図

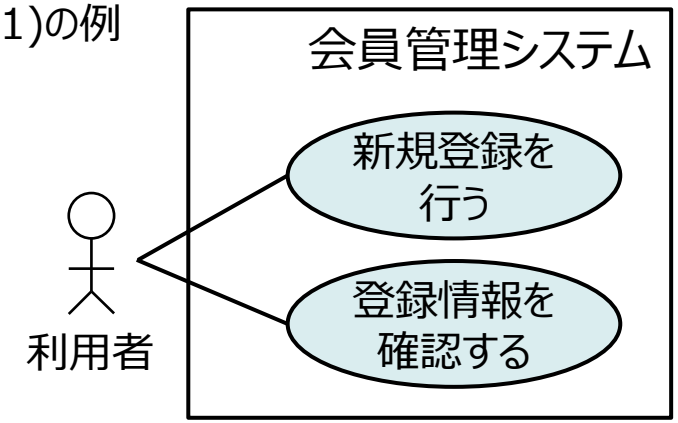
※前のページの概念モデルから作成したものの一部



確認問題

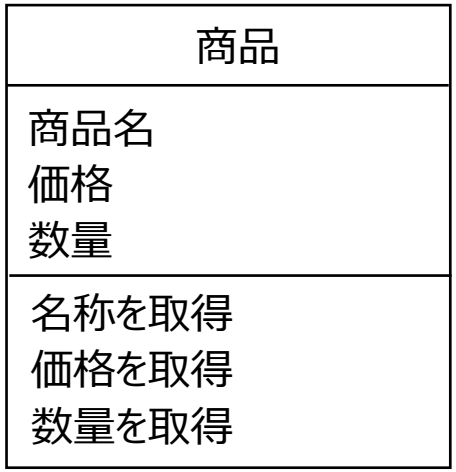
- 次の説明に合うUML図の種類を答えよ。
  - (1) 利用者がどのようにシステムを使用するのかを表す
  - (2) クラスの構造とクラス間の静的な関係を表す
- 以下の各図が示す事柄を説明した文の空欄を埋めよ。

(1)の例



この図は(3)システムの(1)である。
想定するアクタは(4)である。
(4)は(3)システムを用いて、
新規登録、登録情報の確認が可能である。

(2)の例

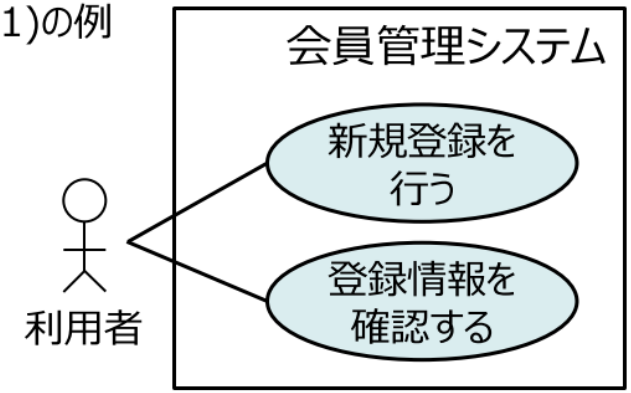


この図は(5)クラスの(2)である。
「商品名」、「価格」、「数量」は、(5)クラスの(6)である。「名称を取得」、「価格を取得」、「数量を取得」は、(5)クラスの(7)である。

確認問題

- 次の説明に合うUML図の種類を答えよ。ユースケース図
  - (1) 利用者がどのようにシステムを使用するのかを表す
  - (2) クラスの構造とクラス間の静的な関係を表す クラス図
- 以下の各図が示す事柄を説明した文の空欄を埋めよ。

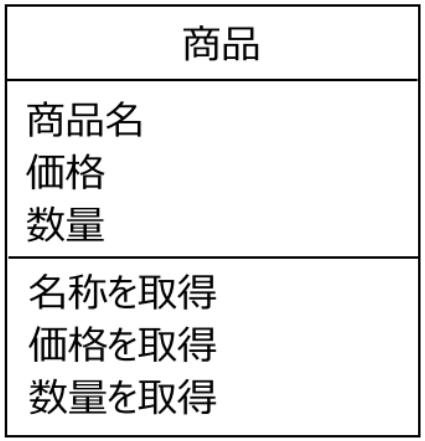
(1)の例



会員管理

この図は(3)システムの(1)である。
想定するアクタは(4)である。利用者
(4)は(3)システムを用いて、
新規登録、登録情報の確認が可能である。

(2)の例



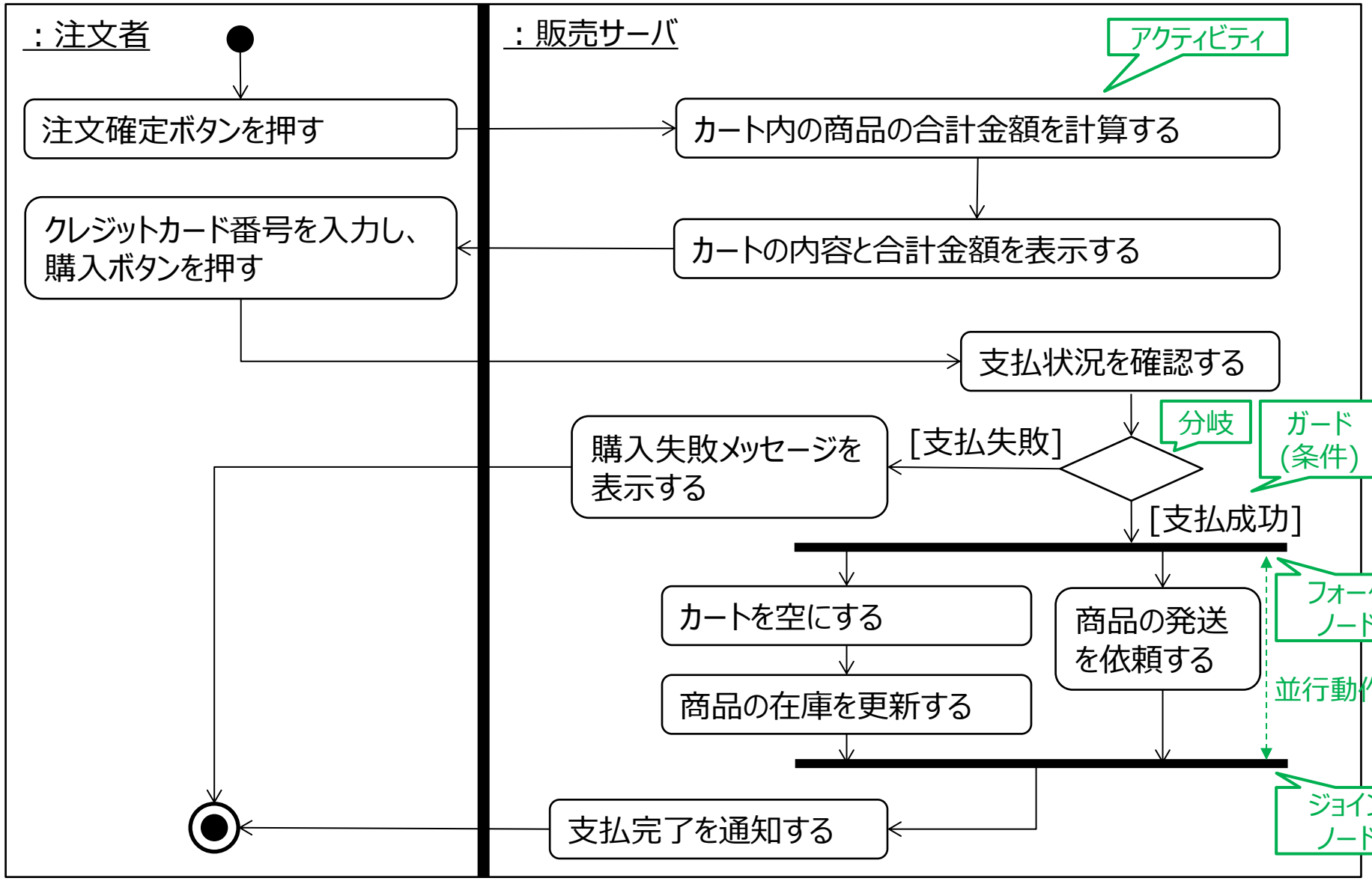
この図は(5)クラスの(2)である。商品
「商品名」、「価格」、「数量」は、(5)クラスの
属性(6)である。「名称を取得」、「価格を取得」、
「数量を取得」は、(5)クラスの(7)である。
振る舞い

振る舞いの記述

- アクティビティ図 activity
  - ある機能を実現するために必要な作業の順序を示す
- シーケンス図 sequence
  - オブジェクト間のメッセージ送受信の時系列
- 状態機械図
  - オブジェクトの状態とイベントによる状態遷移

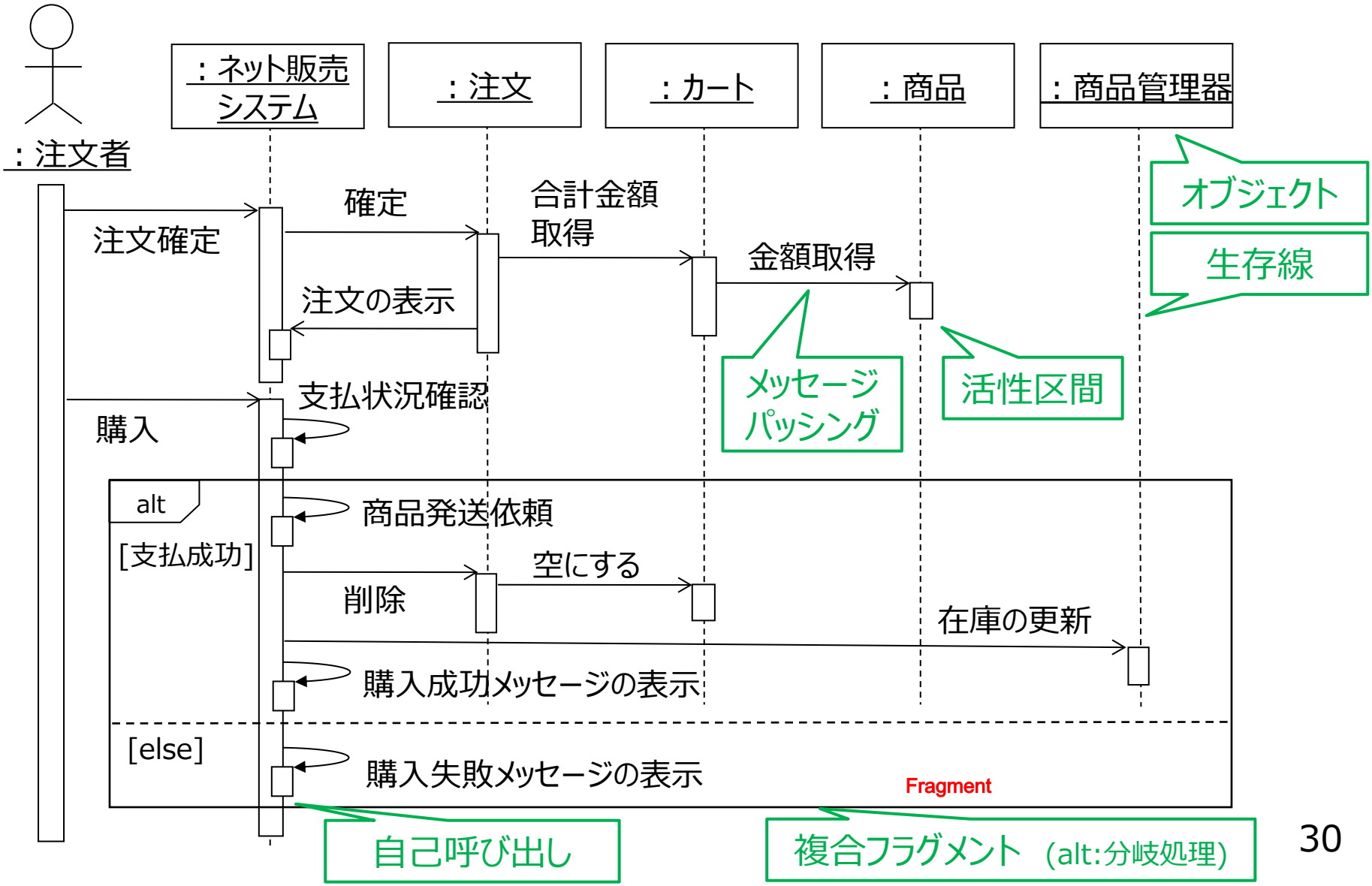
(例) アクティビティ図

ある機能を実現するために
必要な作業の順序を示す



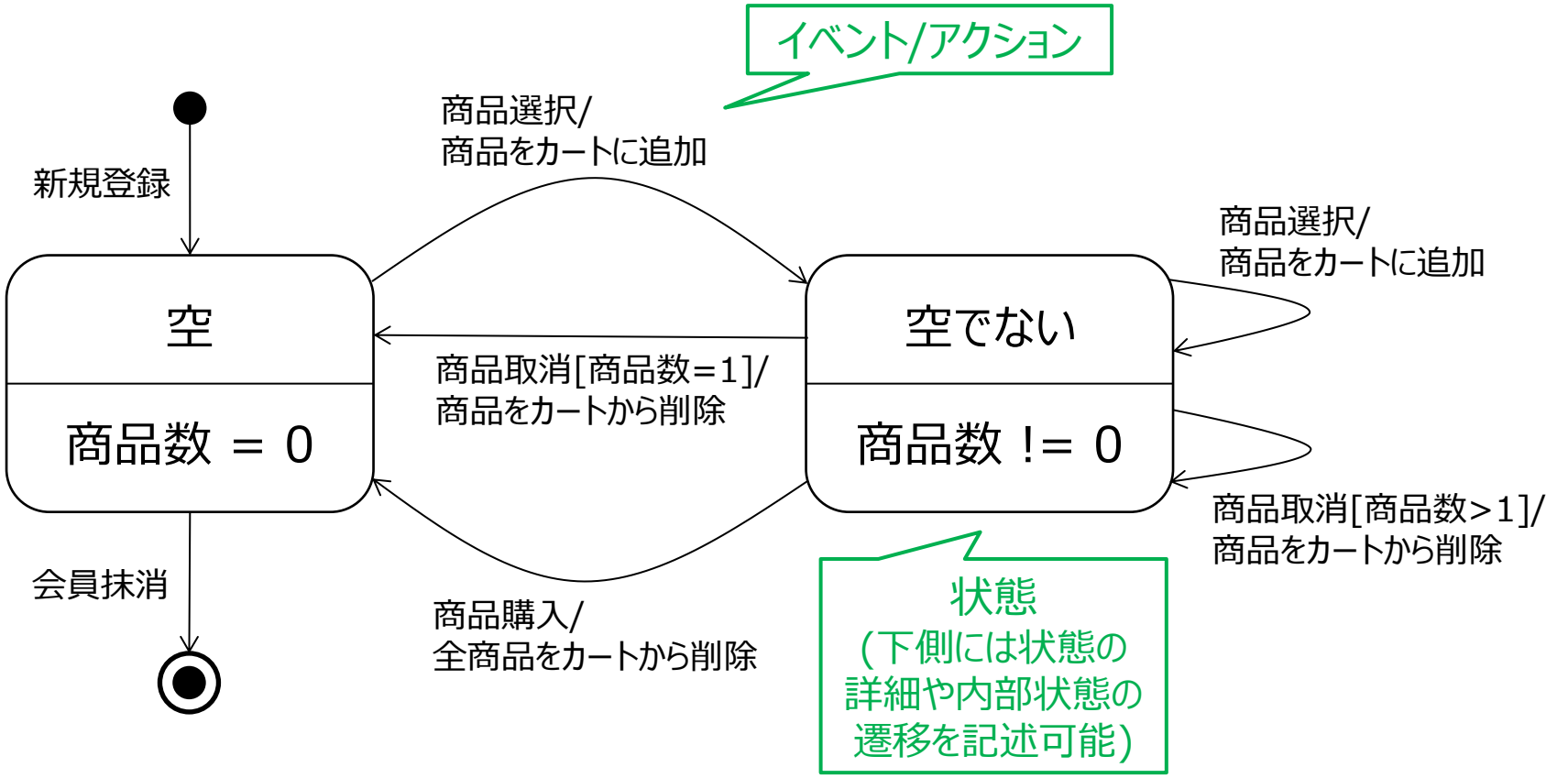
(例) シーケンス図

オブジェクト間のメッセージ送受信の時系列
けいれつ



(例) 状態機械図

オブジェクトの状態と
イベントによる状態遷移



モデルの洗練

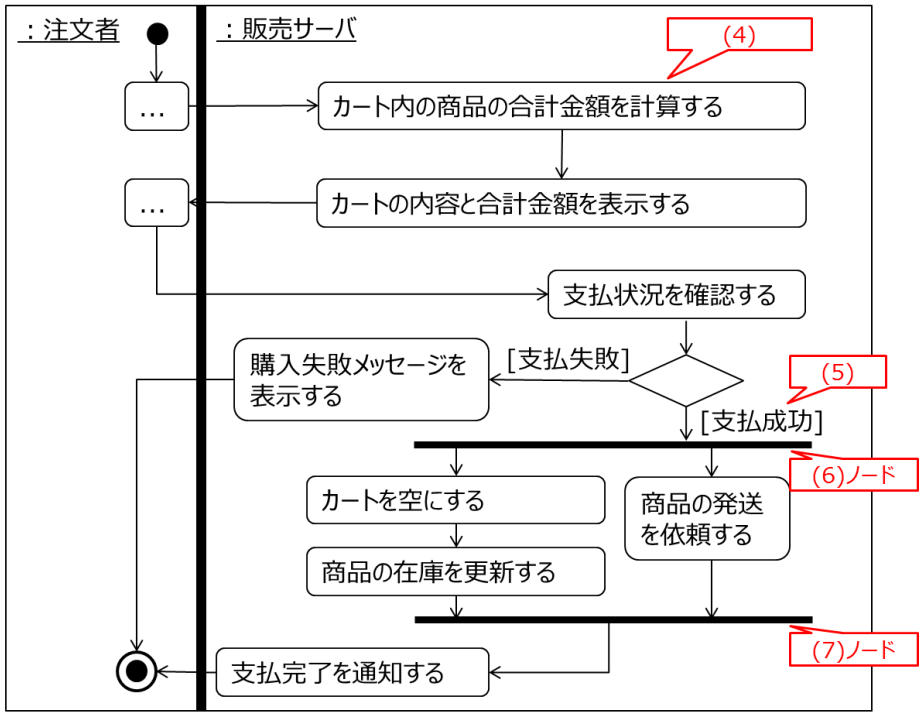
- 振る舞いの分析終了後、再度、クラス図の作成(改善)に戻り、それぞれの図を洗練
- 分析、設計、実装の各段階で必要に応じて図を追加する

- 各図で矛盾がないように注意
  - 各部の対応関係(追跡性)が重要

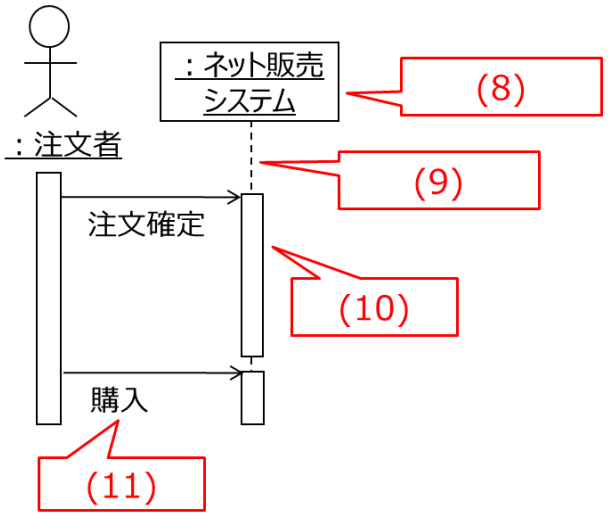
確認問題

- 次の説明に合うUML図の種類を答えよ。
  - (1) ある機能を実現するために必要な作業の順序を示す
  - (2) オブジェクト間のメッセージ送受信の時系列を表す
  - (3) オブジェクトの状態とイベントによる状態遷移を表す
- 以下の各図の該当する部分の名称を答えよ。

(1)の例



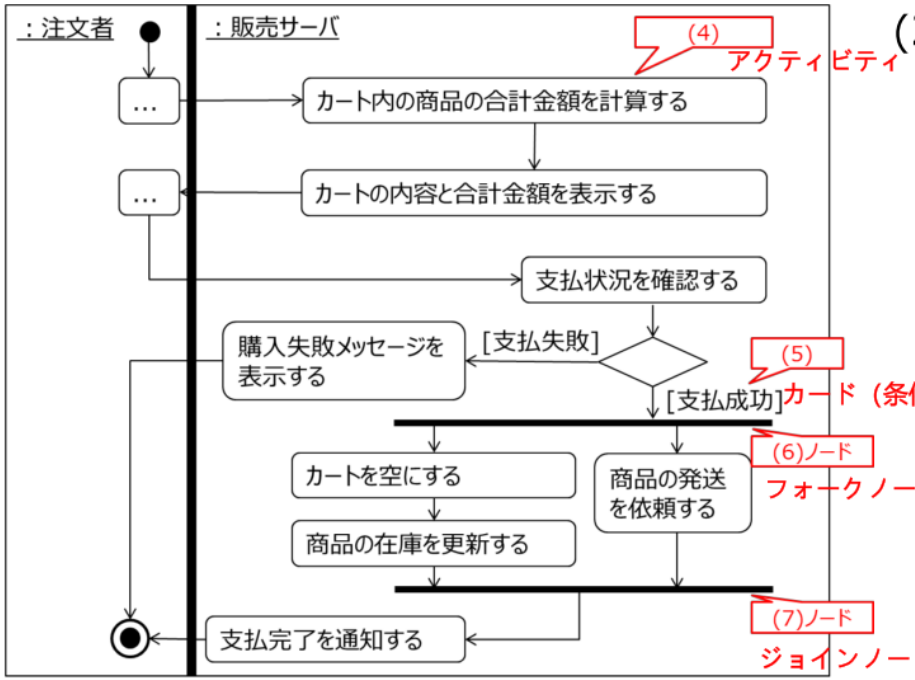
(2)の例



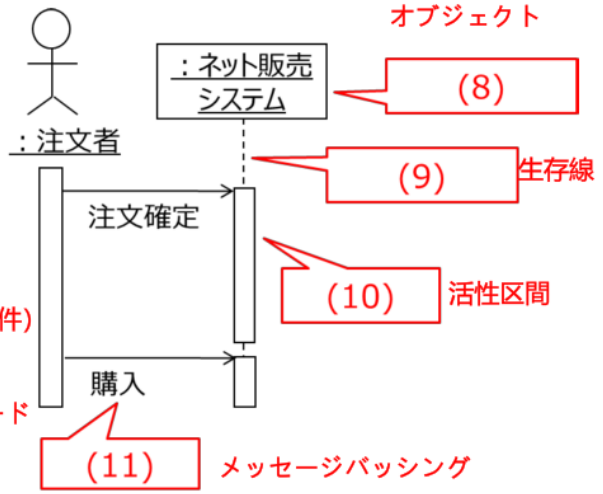
確認問題

- 次の説明に合うUML図の種類を答えよ。アクティビティ図
  - (1) ある機能を実現するために必要な作業の順序を示す
  - (2) オブジェクト間のメッセージ送受信の時系列を表すシーケンス図
  - (3) オブジェクトの状態とイベントによる状態遷移を表す状態機械図
- 以下の各図の該当する部分の名称を答えよ。

(1)の例



(2)の例



参考文献

- 「ソフトウェア工学」
  高橋直久、丸山勝久 著、森北出版、2010
- 「ソフトウェア工学(part2)」
  手塚太郎、http://xi.kc.tsukuba.ac.jp/lec/se2.pdf
- 「OMG Unified Modeling Language Version 2.5」
  Object Management Group, 2015