

情報メディア工学特論

オブジェクト指向システム開発技術

2004/12/7

京都大学高等教育研究開発センター情報メディア教育部門

学術情報メディアセンター連携研究部門(兼任)

大学院工学研究科電気工学専攻(兼担)

小山田耕二

コース概要 (2/2)

- 特異点ベース可視化技術
 - 渦可視化(11/16)
 - 等値面表示高速化、DT-MRI可視化(11/30)
- システム開発技術/OpenGL基礎
 - オブジェクト指向システム開発技術(12/7)
 - 基本オブジェクト設計(12/14)
- 可視化システム実装演習
 - 等値面表示システム(12/21)
 - ボリュームレンダリング表示システム(1/11)
 - 流線表示システム(1/18)

内容

- オブジェクト指向について
- UMLについて
- UMLを使ったシステム開発

オブジェクト指向

- 業務分析から要求定義、設計、プログラミング、開発プロセスまでをカバーする情報システム開発の総合的技術
 - 上流工程における汎用の整理術
 - 従来システム開発の限界を突破した新たな技術
 - 汎化・分類関係・抽象化・is-a関係
 - 集約・所有関係・has-a関係
 - 依存・使用関係・use関係

オブジェクト指向の概念

カプセル化(隠蔽)

オブジェクトの属性や操作を他のオブジェクトから隠す(メリット:仕様変更の最小化)

抽象化(一般化)

より一般的なものに置き換える
(例:14インチブラウン管テレビ テレビ)

インスタンスとクラス

インスタンス:具体的なもの
(例:京大太郎、小泉純一郎)
クラス:抽象化された要素・枠組み
(例:人間)

継承(インヘリタンス)

オブジェクトの属性・操作を引き継ぐ
(例:乗用車・商用車・軍用車 車)

多態性(ポリモフィズム)

異なるオブジェクトに同一の操作を行う
(例:読む 本・新聞・手紙・論文)

UMLとは？

UML (Unified Modeling Language)

UMLは乱立する表記法を統一するために作られた
Unified(統一)という言葉の由来
UMLはオブジェクト指向によるシステム開発で用いられる
さまざまなモデルの表記法を標準化

1997年

OMG (Object Management Group:オブジェクト指向技術の
標準化団体) の標準へ

オブジェクト指向業界での表記法のデファクトスタンダード

UML以前

それぞれ異なる表記法

Booch法

Grady Booch

Grady Booch



1) 参照

オブジェクト
名

オブジェクト
名

オブジェクト
名

OMT法



James
Rumbaugh

2) 参照

オブジェクト

属性

操作

オブジェクト

属性

操作

オブジェクト

属性

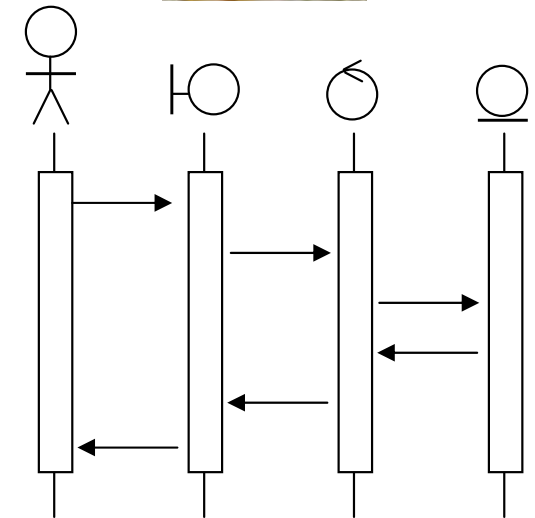
操作

OOSE法

Ivar Jacobson



3) 参照



- 1) <http://www-106.ibm.com/developerworks/library/i-booch/>
- 2) <http://www-306.ibm.com/software/rational/bios/rumbaugh.html>
- 3) <http://www.jaczone.com/postcards/>

UMLの歴史

1994	BoochとRumbaughがモデリング技法統一へ
1995	Jacobsonが統一作業に参加
1996	UML0.9
1997(Jan)	UML1.0
1997(Sep)	UML1.1
1997(Nov)	UML1.1がOMG標準に
1998	UML1.2
1999	UML1.3
2001	UML1.4
2003	UML1.5
2003	UML2.0ドラフト公開

OMG : Object Management Group (オブジェクト指向技術標準化団体)

UMLのダイアグラム

UMLの主なダイアグラム

ユースケース図 システムがどのように機能すべきか(ユースケース)、およびその外部環境(アクター)を表す

クラス図 システムの静的な構造を表現

オブジェクト図 システムのある時点における静的な構造を表現

相互作用図 オブジェクト間の相互作用を表現
(シーケンス図・コラボレーション図)

状態チャート図 オブジェクトの状態遷移を表現

アクティビティ図 処理や業務の流れを表現

コンポーネント図 コンポーネント間の依存関係を表現

配置図 システムの物理的な構成を表現

パッケージ図 パッケージ間の依存関係を表現

モデリング

モデル: ある対象を分析して整理し、表現したもの
モデリング: モデルを作成する作業

Step1: 要求モデリング

ユーザの要求把握 (システム化の対象範囲を明確に)

Step2: 分析モデリング

システム化の対象整理 (システムの構造を明確に)

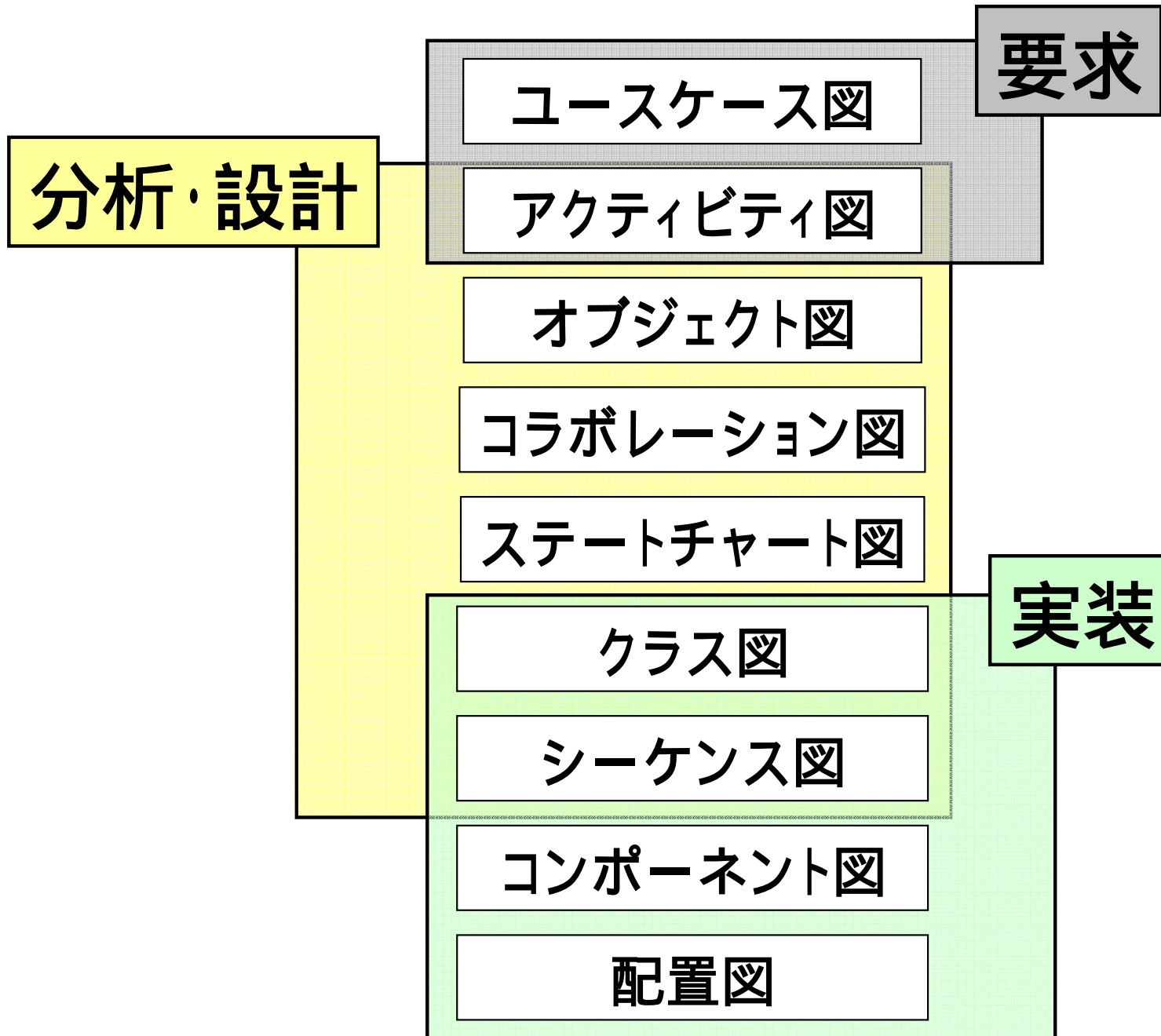
Step3: 設計モデリング

システム化の実現方法定義 (システム内部の仕様設計)

Step4: 実装モデリング

システムの構成要素定義 (構成・配置・動作を記述)

設計工程とUML



ユースケース図(1)

ユースケース図を使ったモデリング

- ・システムがどのように機能すべきか(ユースケース)、およびその外部環境(アクター)を表す図
- ・エンドユーザの視点からシステムを見る
- ・システムの外部と内部との境界を明確化する

「世界中の酒」の通信販売を行っている兄弟社は、昨今の焼酎ブームを当て込み個人向け販売情報システムを導入することになった。
販売業務のみをシステム化の範囲とする。

ユースケース図(2)

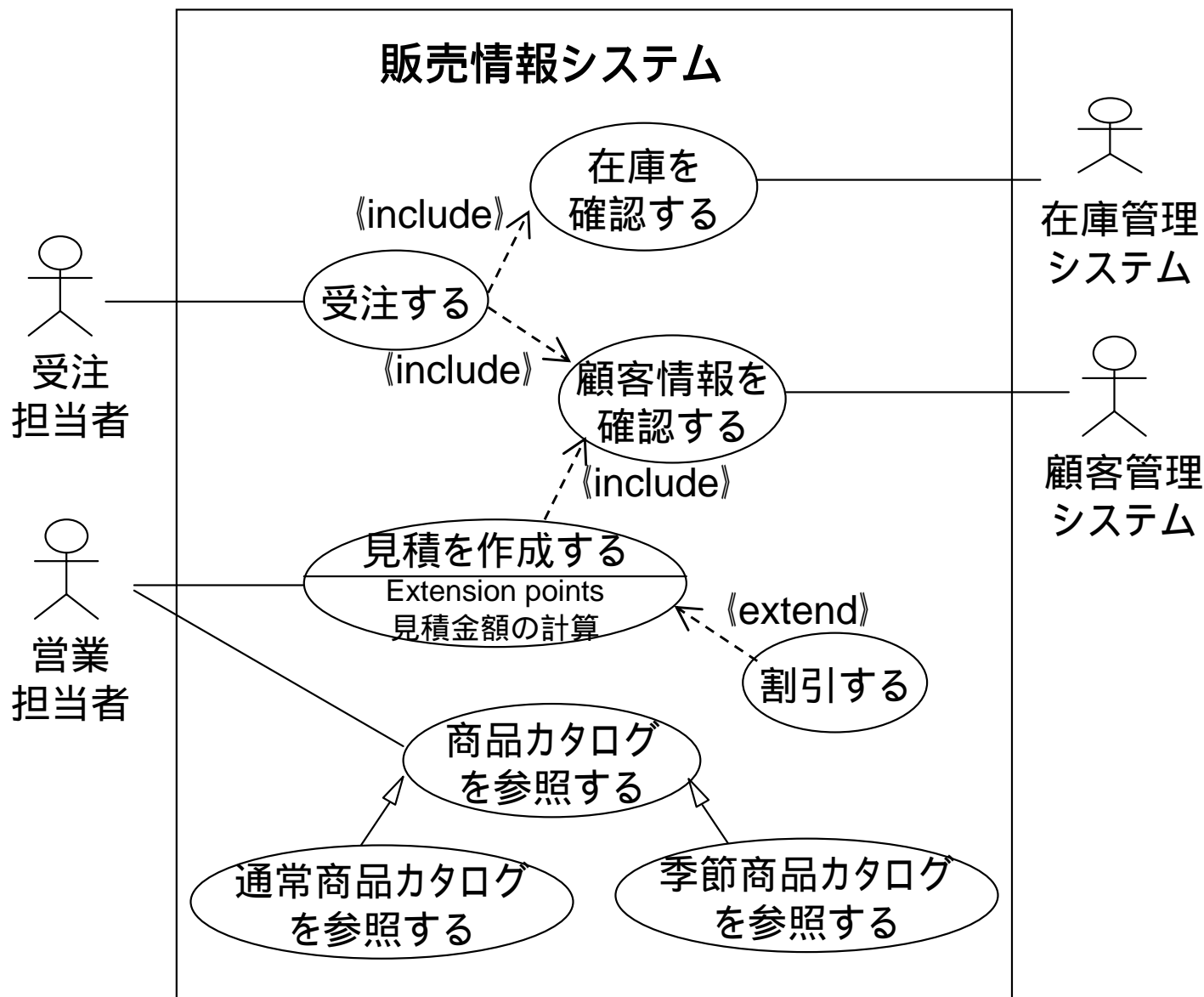
シナリオ

受注担当者が受注すると本システムでは、在庫管理システムや顧客管理システムを使って、在庫や顧客情報を確認する。

営業担当者の場合は、さらに見積もりを作成したり、商品カタログを参照する。商品カタログでは、通常商品と季節商品向けのカタログが用紙されている。見積もり作成では、顧客管理システムを使って、顧客情報を確認し、場合によっては、割引を行う。

仕入れ担当者は、仕入れするにあたって、在庫管理システム、売上げ情報システムや価格情報システムを使って、在庫、売上げ情報や価格情報を確認する。

ユースケース図(3)



ユースケース図(4)

ユースケース

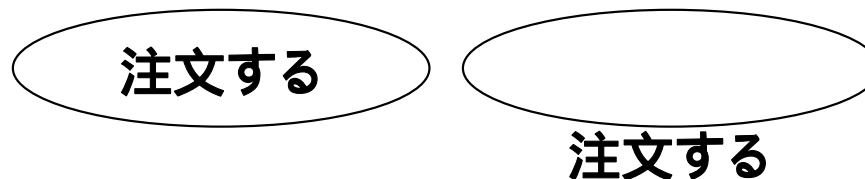
ユーザなどシステム外部から見たシステムの振る舞いを表す。システムの振る舞いとはシステムがどのように動作し、反応するかということ

アクター

システムと相互作用する外部の実体を抽象化したもの

表記法

楕円で表し、内側または下にユースケースを記述例)

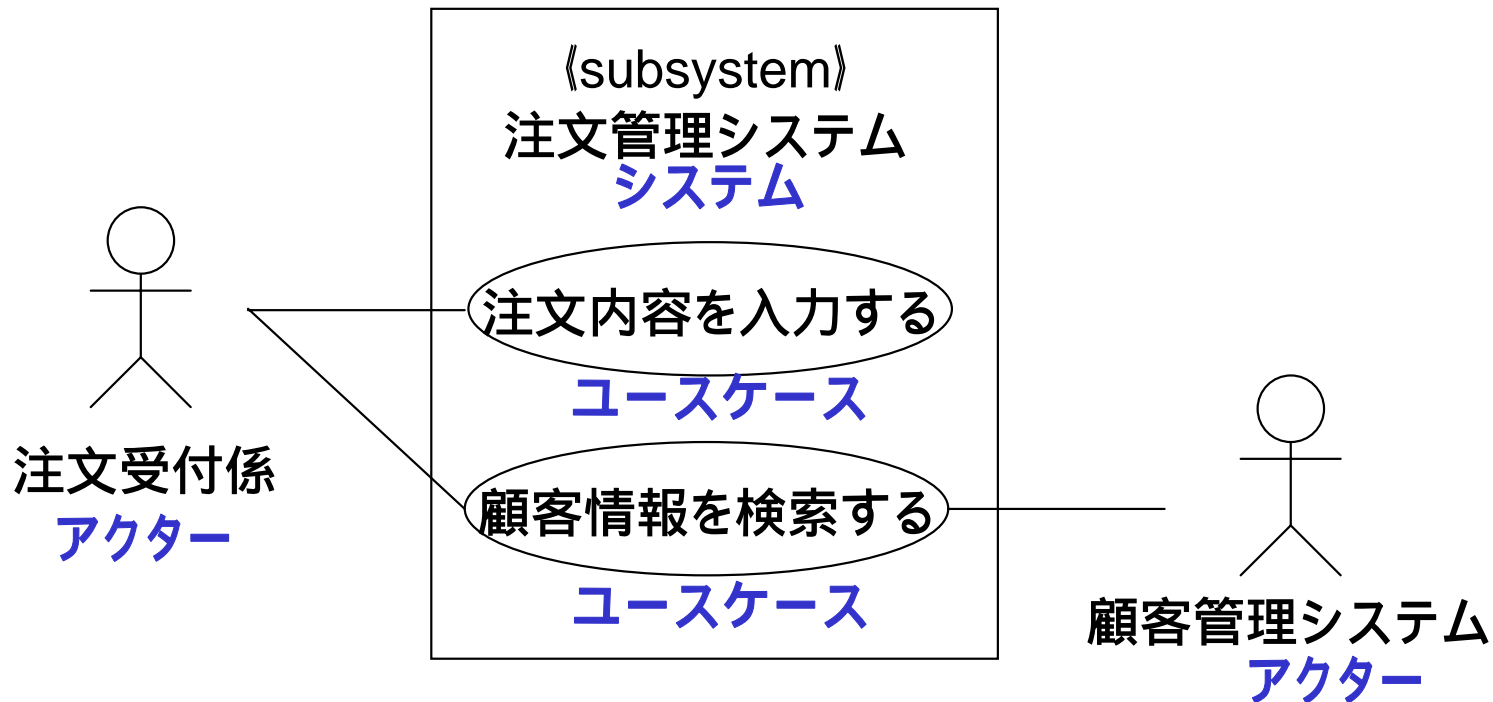


ユースケース図(5)

ユースケース図

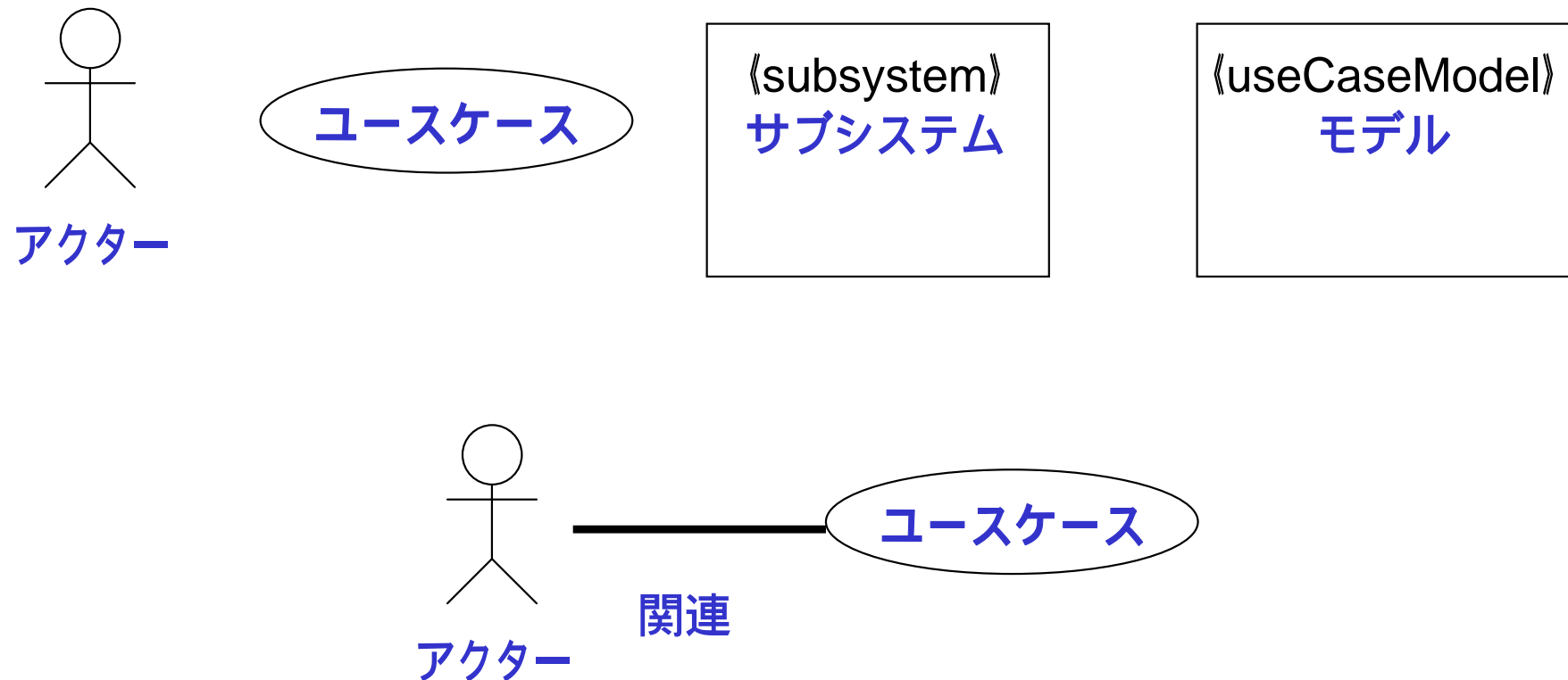
システムが提供する機能とそれに関連する外部要素
(ユーザなど)を表す

例)「注文管理システム」:「注文内容入力」と「顧客情報検索」を提供
ユーザは「注文受付係」/「顧客管理システム」を利用



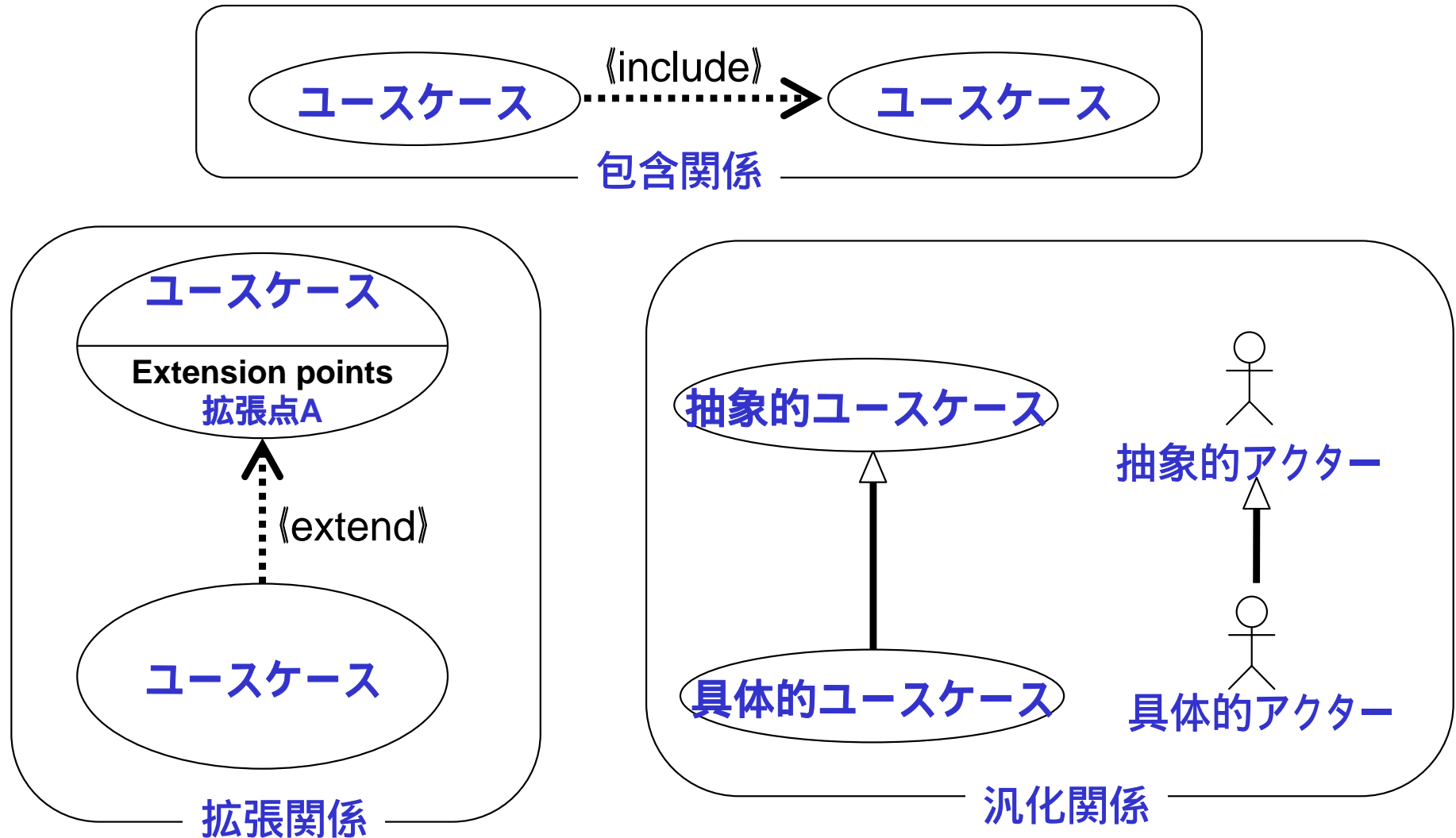
ユースケース図(6)

ユースケース図の要素(1)



ユースケース図(7)

ユースケース図の要素(2)



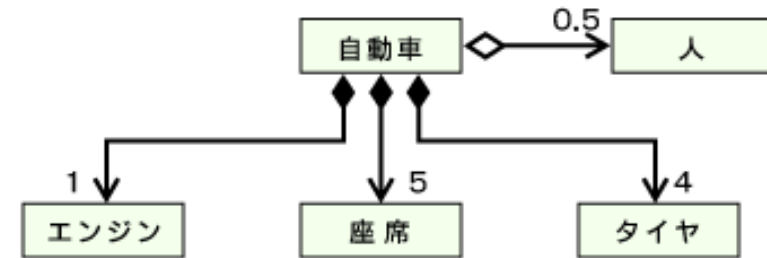
クラス図とオブジェクト図

- クラス図

- モデルの静的構造を表す
- システムの機能に着目して、問題領域を論理的、静的に見る
- 設計段階では、開発者の視点からモデルが詳細化される
- クラス図の情報が実装言語のクラスに対応付けられる

- オブジェクト図

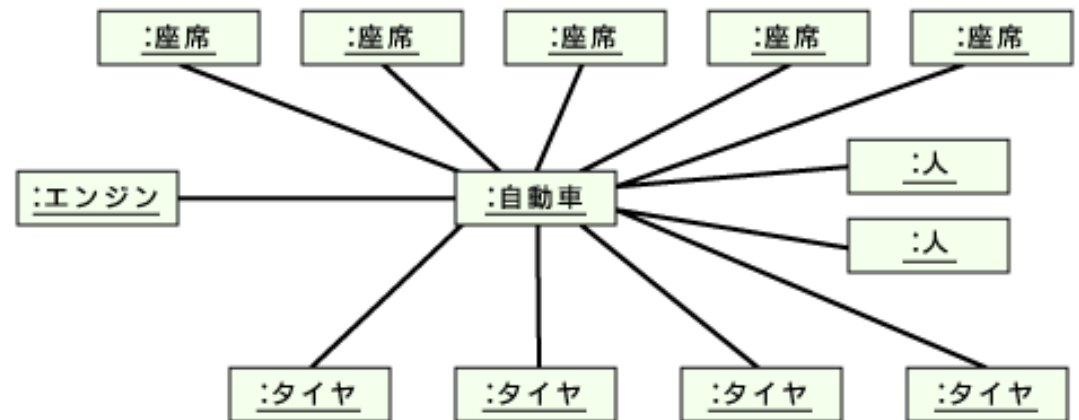
- クラス図を実体化したもの



【クラス図】



【オブジェクト図】



オブジェクト図(1)

受注関連部分のモデル化

常連客であり、酒の小売業を営むAさん(顧客番号:034、住所:京都市左京区吉田XXX、電話番号:075-753-XXXX)から、本日以下受注明細で示される商品を受注した。

焼酎「キョウダイ」(商品コード001、単価980円):200本

ワイン「ブジョレー」(商品コード026、単価1280円):300本

日本酒「寒ばしり」(商品コード003、単価4280円):50本

常連客であるAさんの焼酎、ワイン、日本酒の割引率はそれぞれ15%、20%、10%である。
業務概要をクラス図やオブジェクト図を用いてモデル化する。

(ヒント)

「Aさん:顧客」、「Aさんからの受注:受注」、

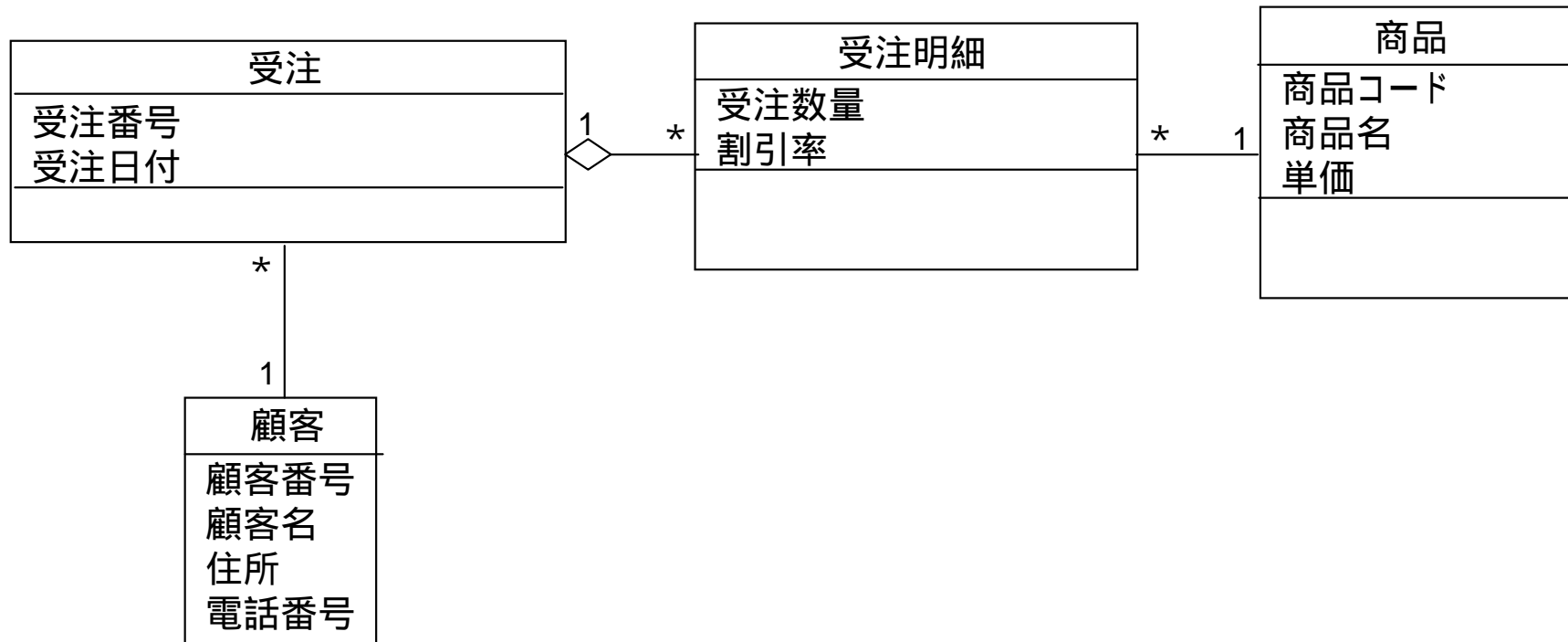
「焼酎の受注明細:受注明細」、「ワインの受注明細:受注明細」、

「日本酒の受注明細:受注明細」、

「焼酎:商品」、「ワイン:商品」、「日本酒:商品」

これらの関係を考える

クラス図(1)



クラス図(2)

クラス

オブジェクトの共通的要素を抽象化し、それを枠組みとして定義したもの
クラスとオブジェクトは「雛形」と「その実体」という関係

表記方法

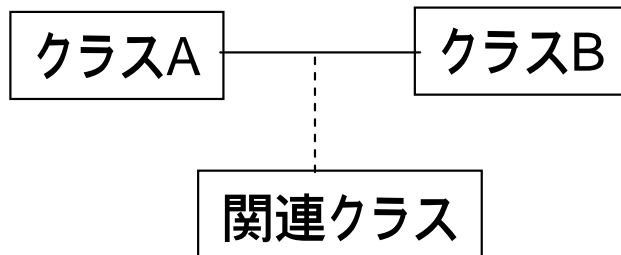
クラス名
属性1:型 = 初期値 属性2:[多重度 ordered]
操作(パラメータ:型 = デフォルト値):戻り値の型

クラス図(3)

クラス図の要素(1)

クラス
+ 属性 (public)
属性 (protected)
- 属性 (private)
~ 属性 (package)
+ 操作 (public) ()
操作 (protected) ()
- 操作 (private) ()
~ 操作 (package) ()

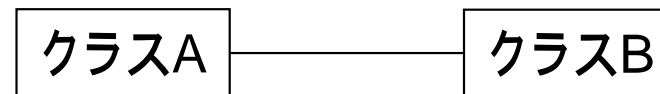
可視性



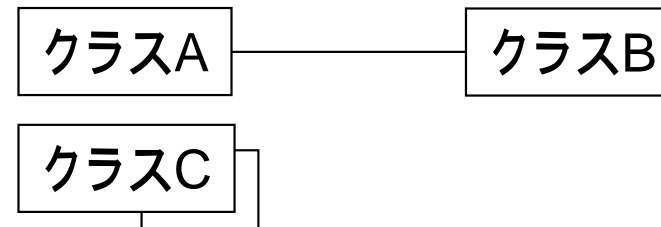
関連クラス

クラス
<u>属性(クラススコープ)</u>
属性(インスタンススコープ)
<u>操作(クラススコープ)()</u>
操作(インスタンススコープ)()

スコープ



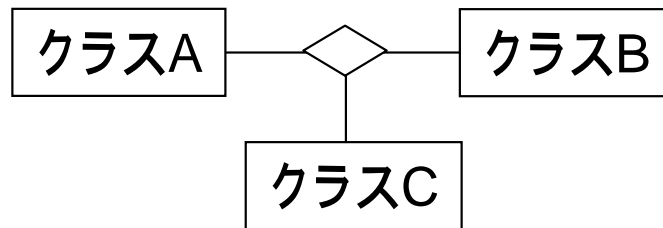
関連



二項関連

クラス図(4)

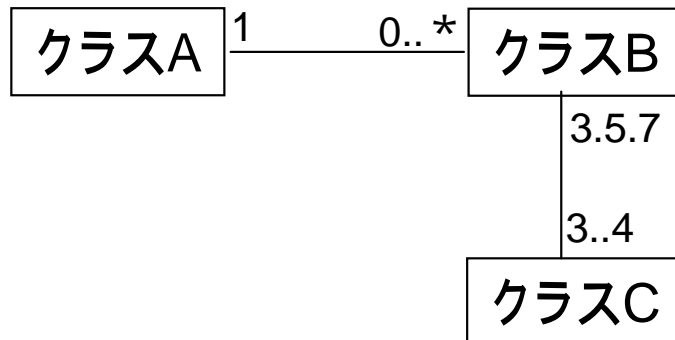
クラス図の要素(2)



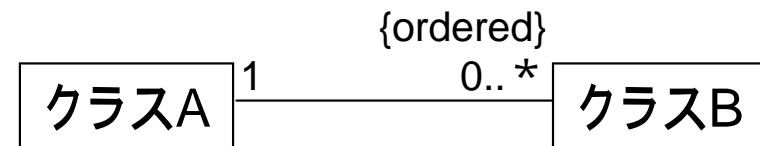
N項関連



関連名



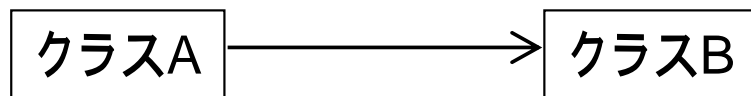
多重度



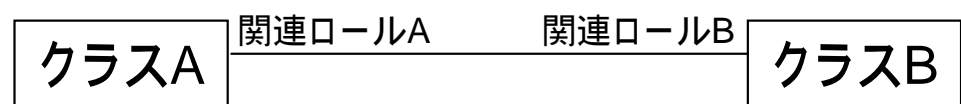
順位付け

クラス図(5)

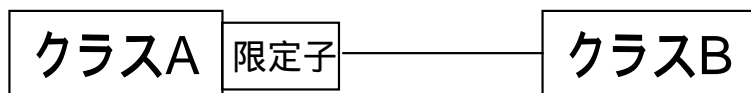
クラス図の要素(3)



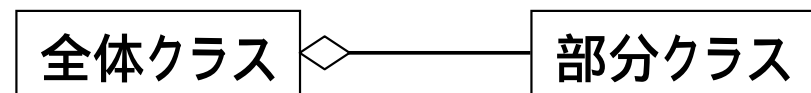
誘導可能性



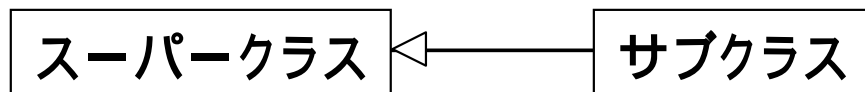
関連ロール



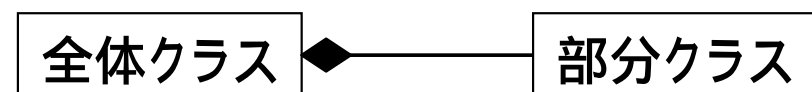
限定子



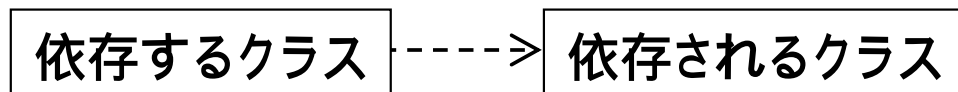
集約



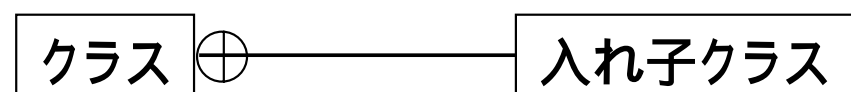
汎化



コンポジション



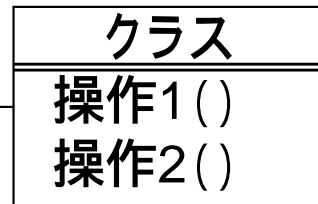
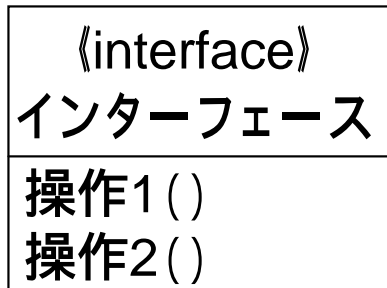
依存



入れ子クラス

クラス図 (6)

クラス図の要素 (4)

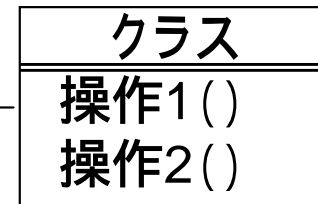


抽象クラス

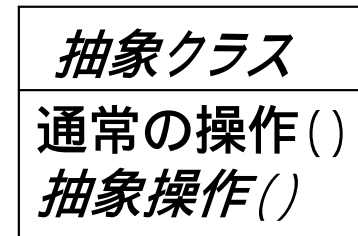
抽象クラス
{abstract}

抽象クラス

インターフェース
(ロリポップ表記)



実現



抽象操作



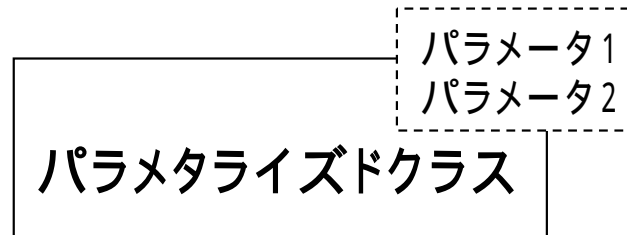
タイプクラスと実装クラス



ユーティリティクラス



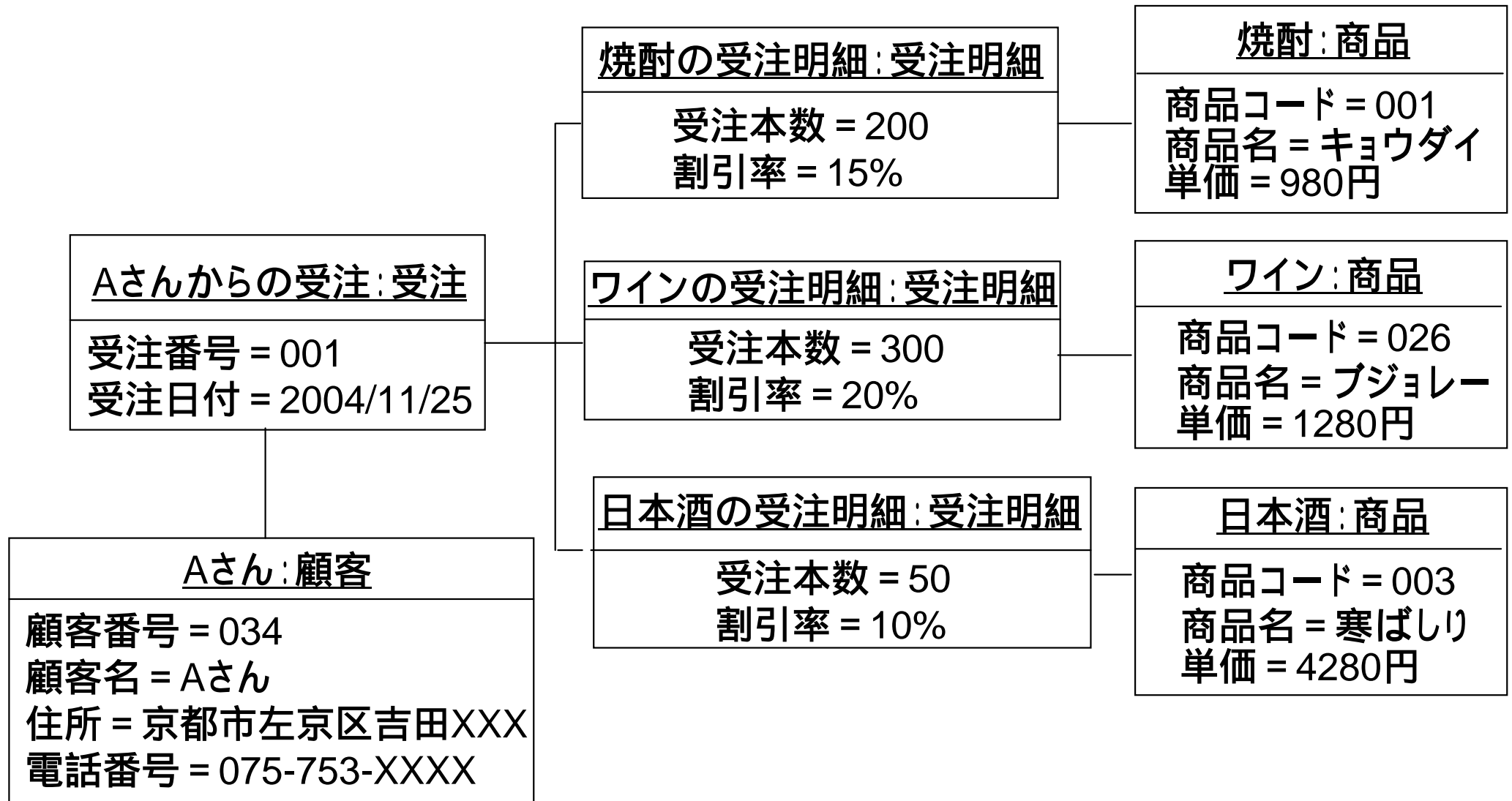
メタクラス



パラメタライズドクラス

オブジェクト図(1)

クラス図の実体化



オブジェクト図(2)

オブジェクト

「具体的なモノ」のこと
(人、車、PC、情報など現実世界にあるモノ)
オブジェクト名、属性値、状態で構成される

表記法

四角の箱で表し、以下のように記述

<u>オブジェクト名 / ロール : クラス名</u> [状態名]

属性名 = 値

例)

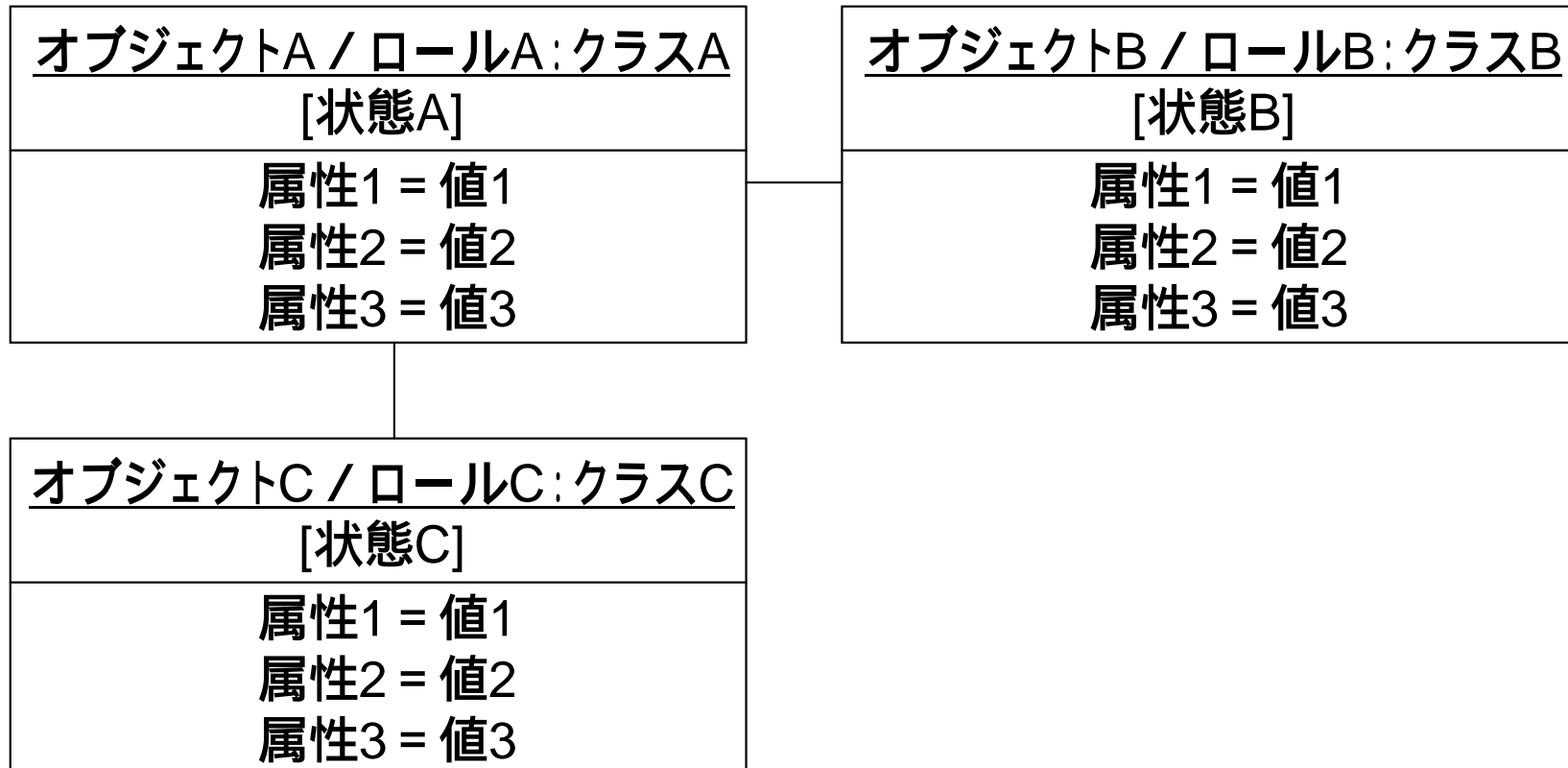
<u>オブジェクト名 / ロール : クラス名</u> [状態名]

属性名 = 値

オブジェクト図(3)

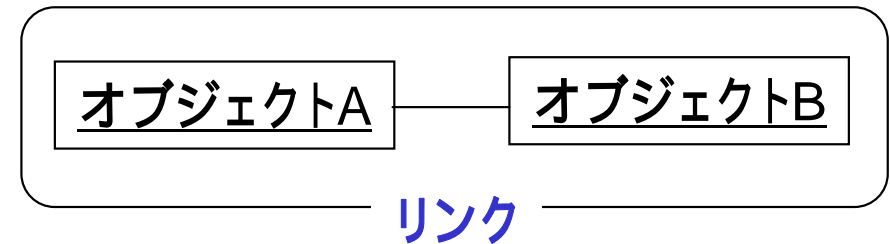
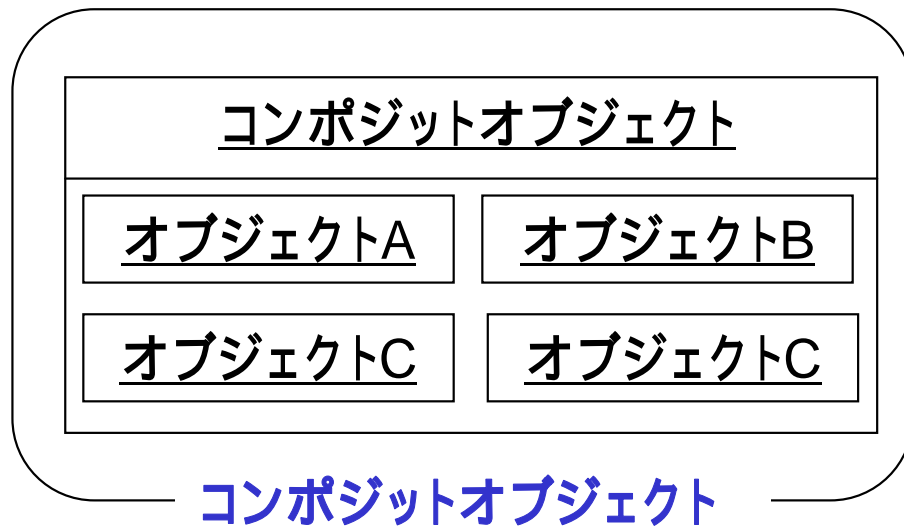
オブジェクト図

オブジェクト間の静的な関係を把握するために作成
システムのある時点におけるスナップショットを表す



オブジェクト図(4)

オブジェクト図の要素



オブジェクト間の相互作用

オブジェクトの振る舞いを抽出する

受注係は、Cさんからの注文「焼酎キョウダイ200本」に応じて、

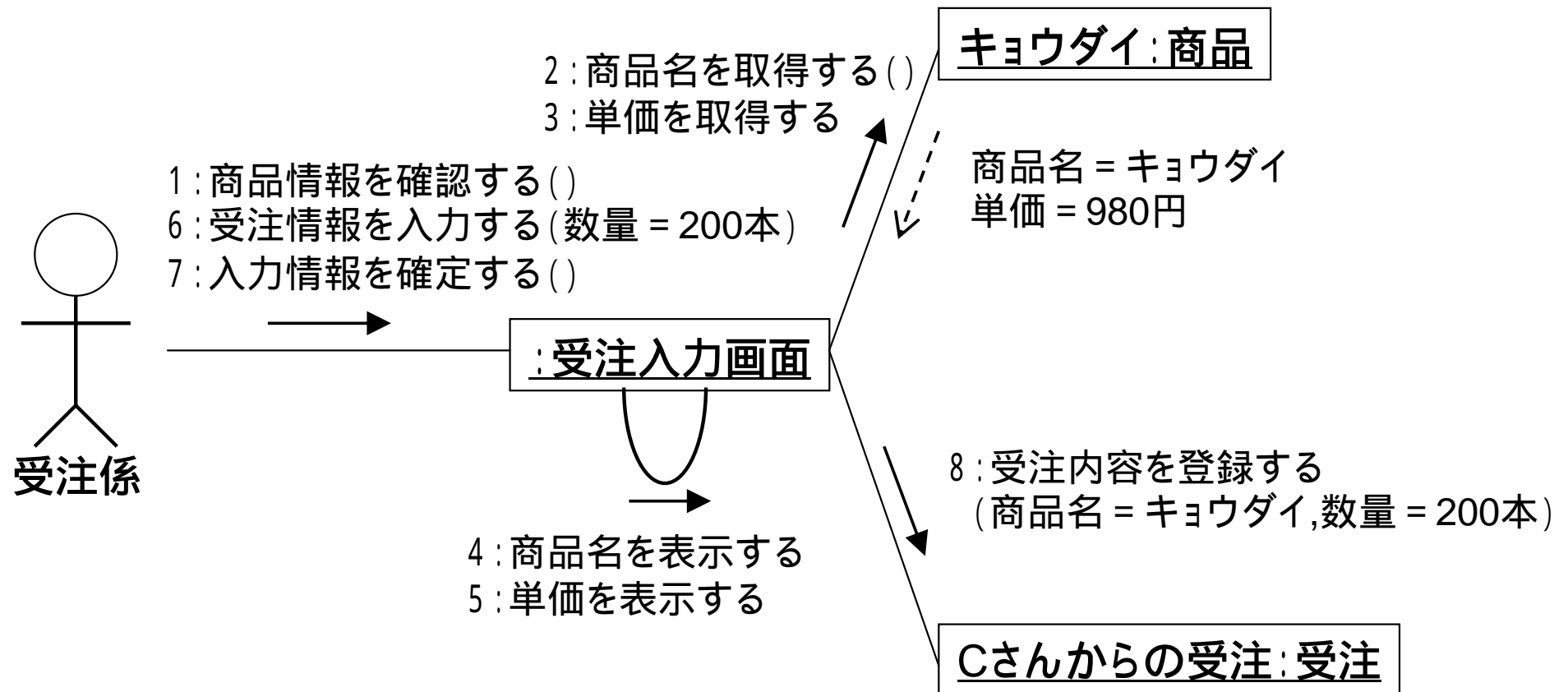
1. 入力画面に対して商品情報を確認する
2. 商品情報から商品名を取得する
3. 商品情報から商品名「キョウダイ」が入力画面に出される
4. 単価情報を商品情報から取得する
5. 単価情報が入力画面に返される
6. 受注入力画面に商品情報を出力する(商品名、単価)
7. 入力画面に対して受注情報(本数)を入力する
8. 入力情報を確定する
9. 入力画面から受注内容をBさんからの注文に登録する

(ヒント)

「受注係」、「受注入力画面」、「商品」、「受注」の関係を考える

コラボレーション図(1)

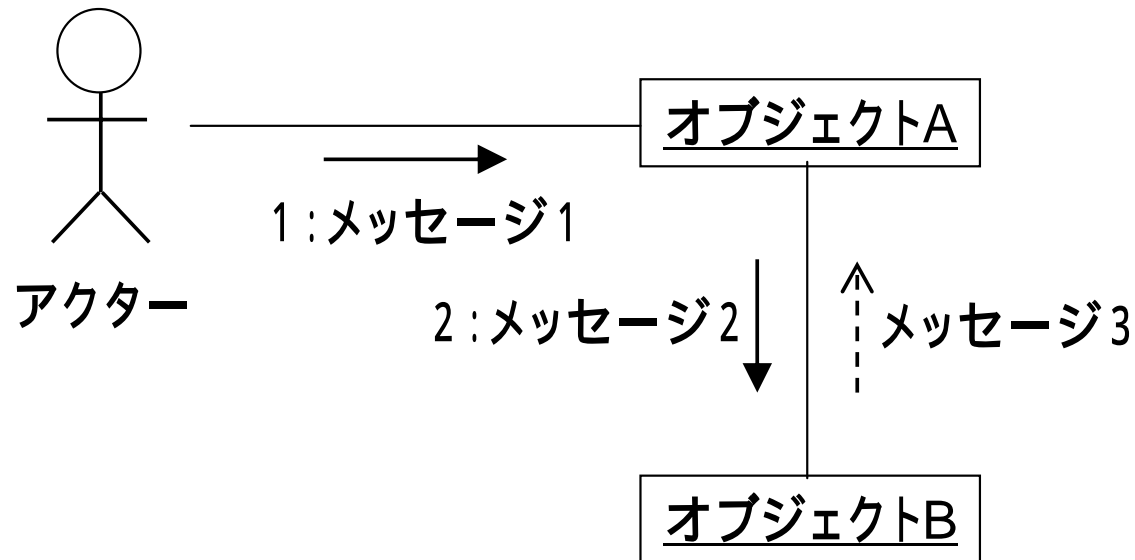
オブジェクト間の相互作用を表現したもの。



コラボレーション図(2)

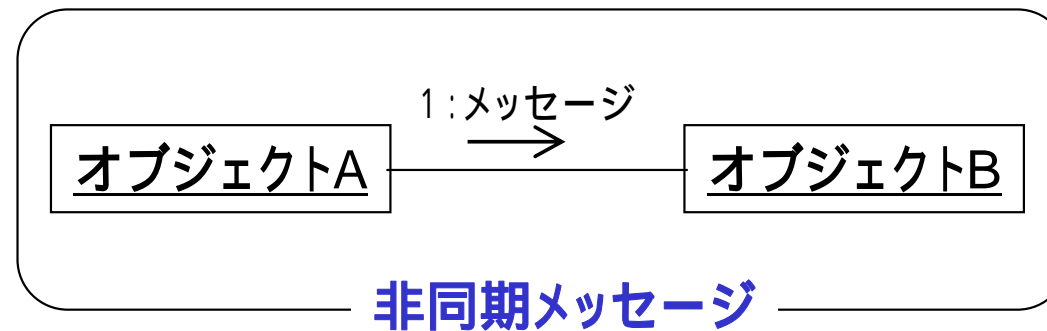
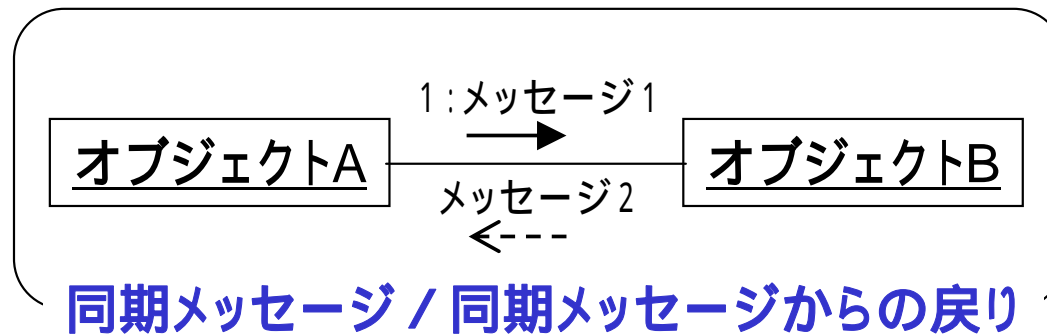
コラボレーション図

「相互作用図」の一種
オブジェクト間の関係を重視する



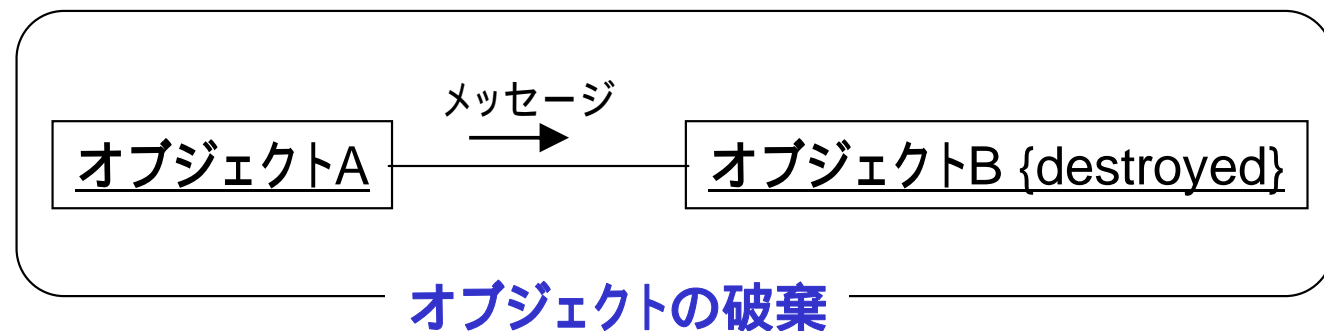
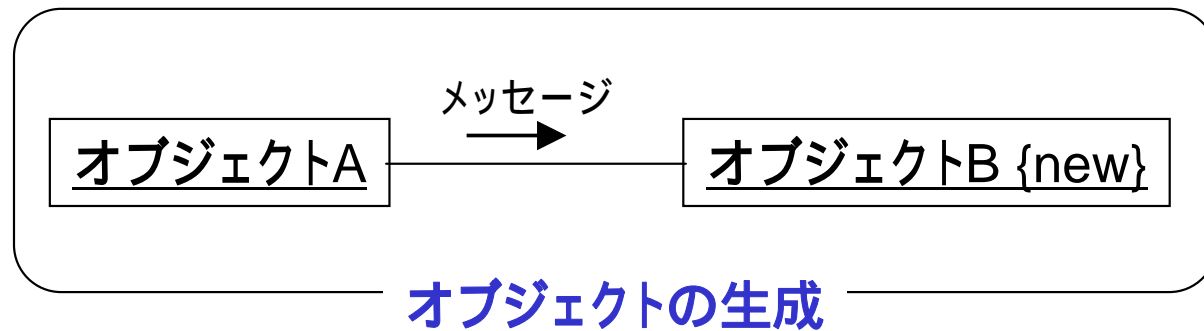
コラボレーション図(3)

コラボレーション図の要素(1)



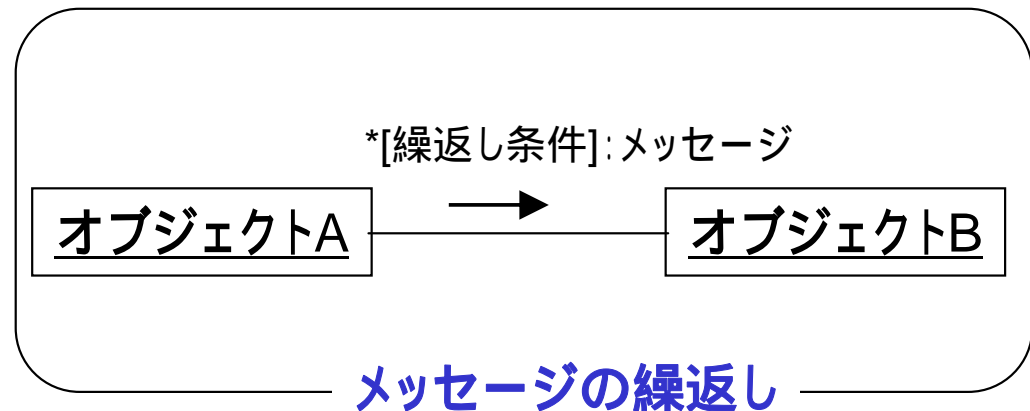
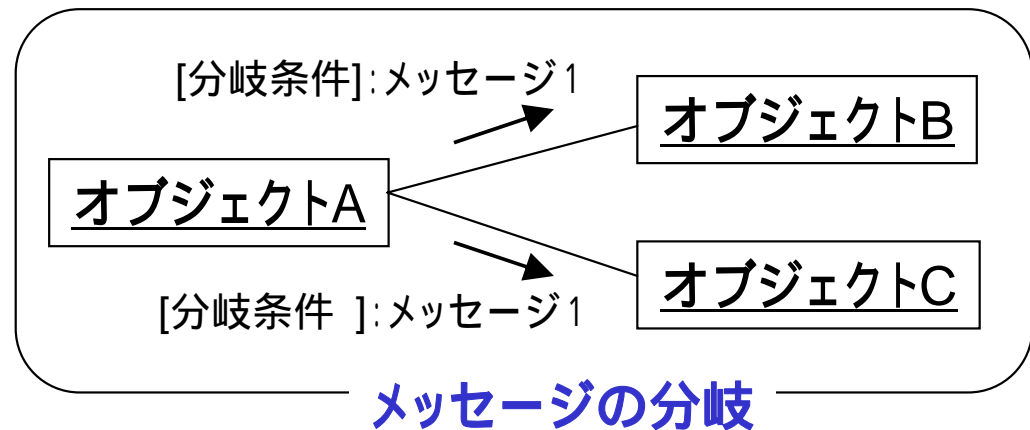
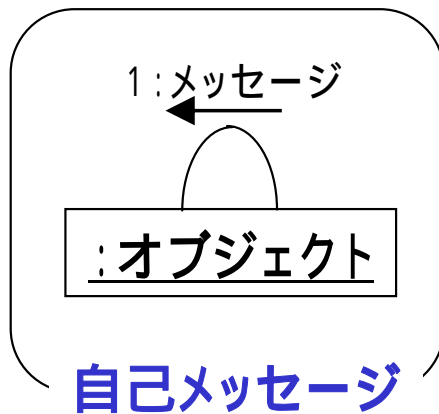
コラボレーション図(4)

コラボレーション図の要素(2)



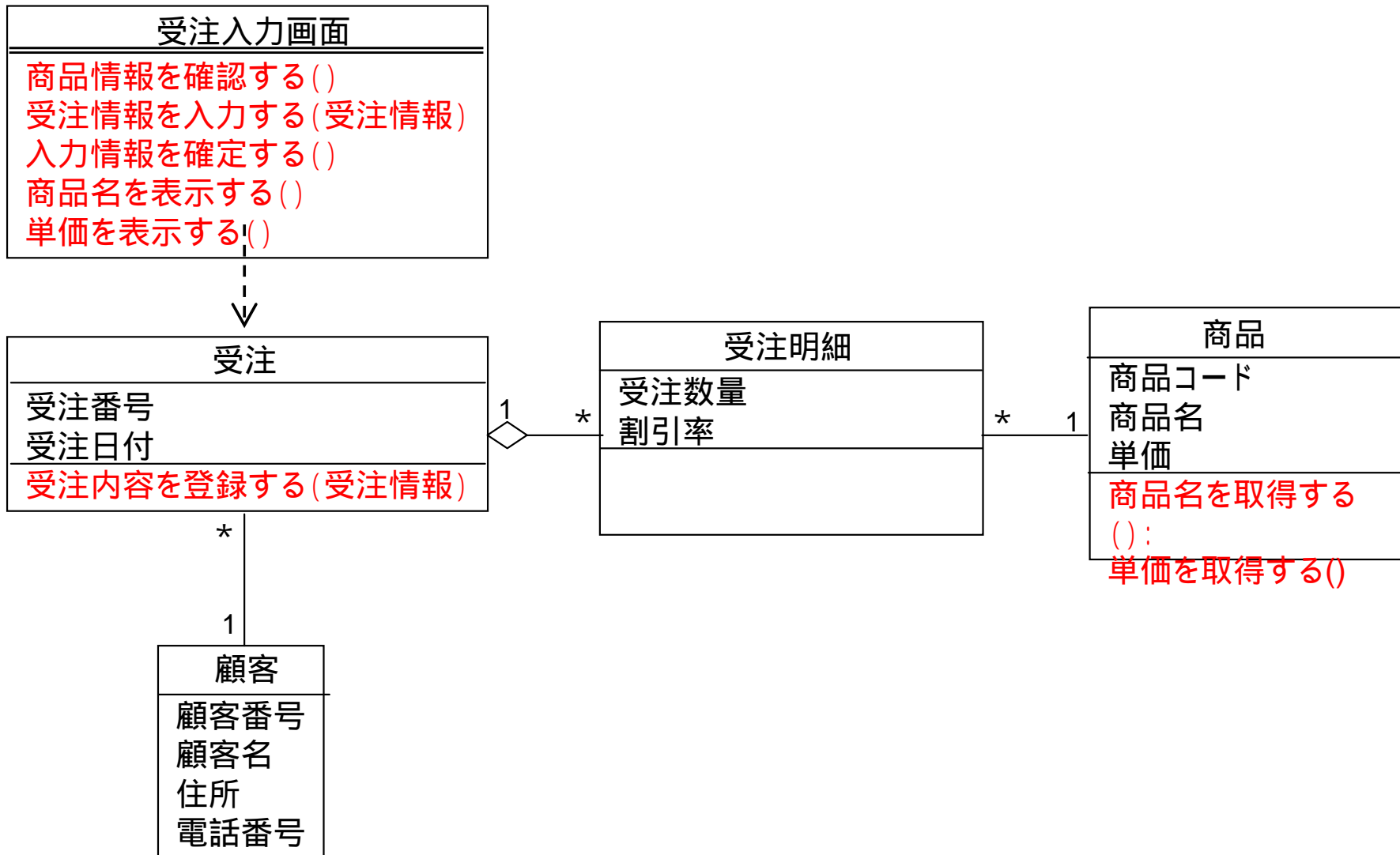
コラボレーション図(5)

コラボレーション図の要素(3)



クラス図の洗練(1)

抽出された振る舞いを操作として記述



相互作用の追加(1)

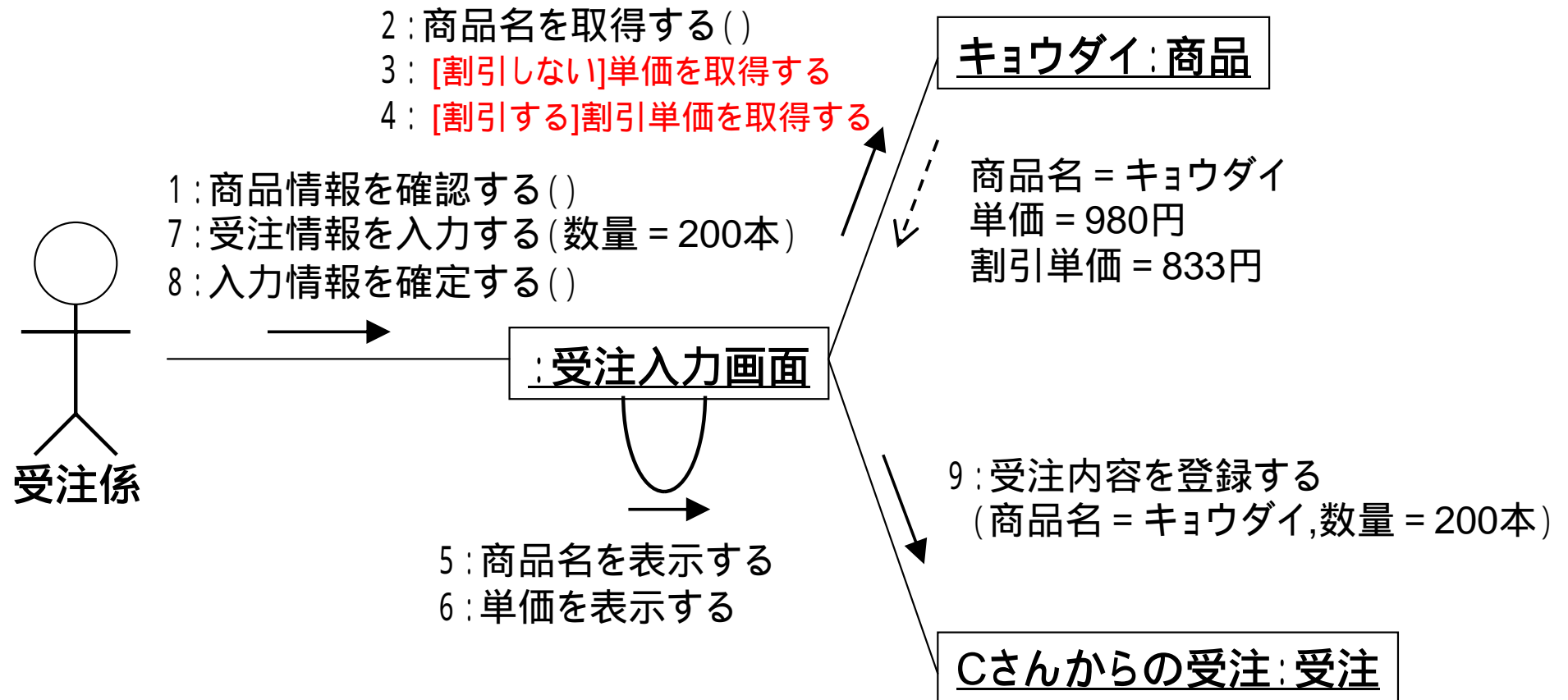
オブジェクトの振る舞いの追加

Bさんから焼酎キョウダイに対して200本の注文があった
各オブジェクト間のやり取りは以下のようなものとする
オブジェクトは受注係、受注入力画面、商品情報、Bさんからの注文とする

1. 入力画面に対して商品情報を確認する
2. 商品情報から商品名を取得する
3. 商品情報から商品名「キョウダイ」が入力画面に出される
4. 単価情報を商品情報から取得する
5. ・Bさんが会員の場合、単価は833円と入力画面に返される
6. ・Bさんが会員でない場合、単価は980円と入力画面に返される
7. 受注入力画面に商品情報を出力する(商品名、単価)
8. 入力画面に対して受注情報(本数)を入力する
9. 入力情報を確定する
10. 入力画面から受注内容をBさんからの注文に登録する

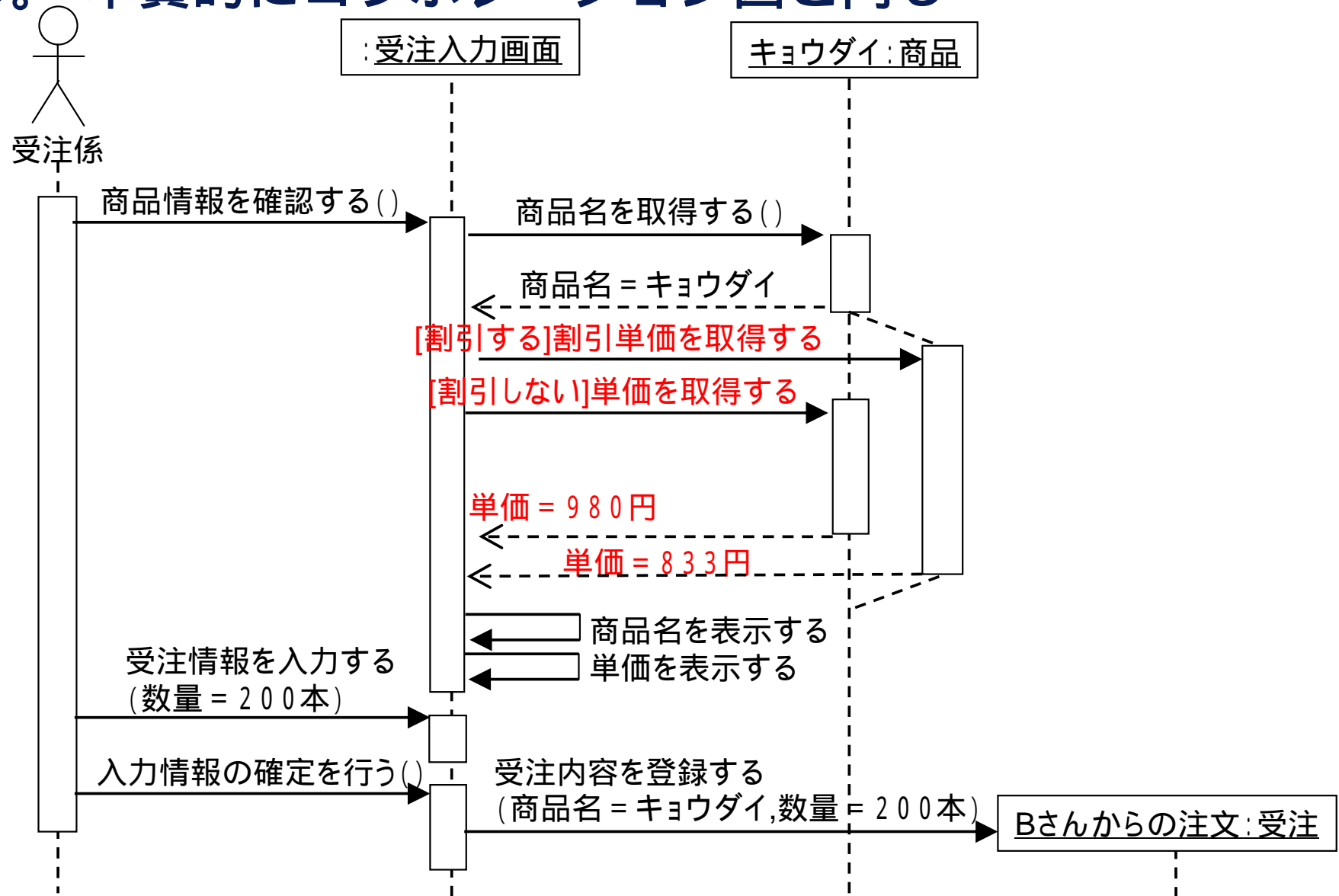
コラボレーション図(6)

割引の有無を考慮



シーケンス図(1)

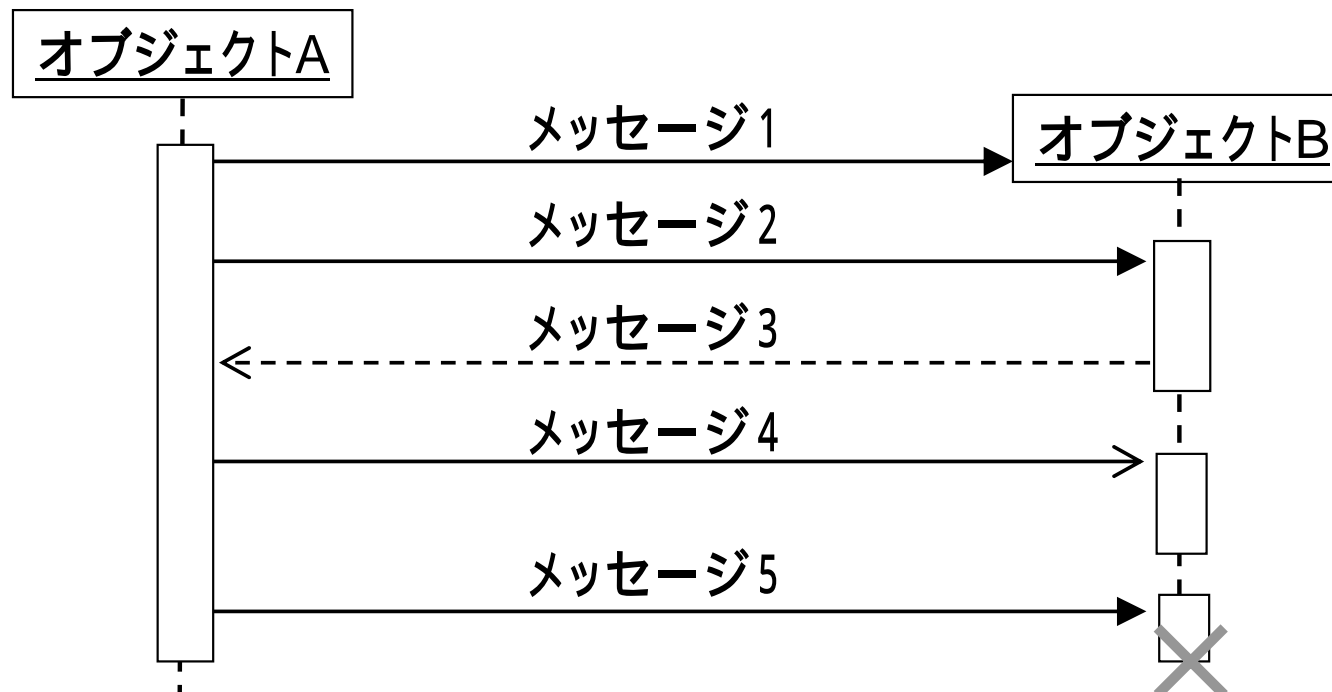
オブジェクト間の相互作用を時間に沿って表現したもの。本質的にコラボレーション図と同じ



シーケンス図(2)

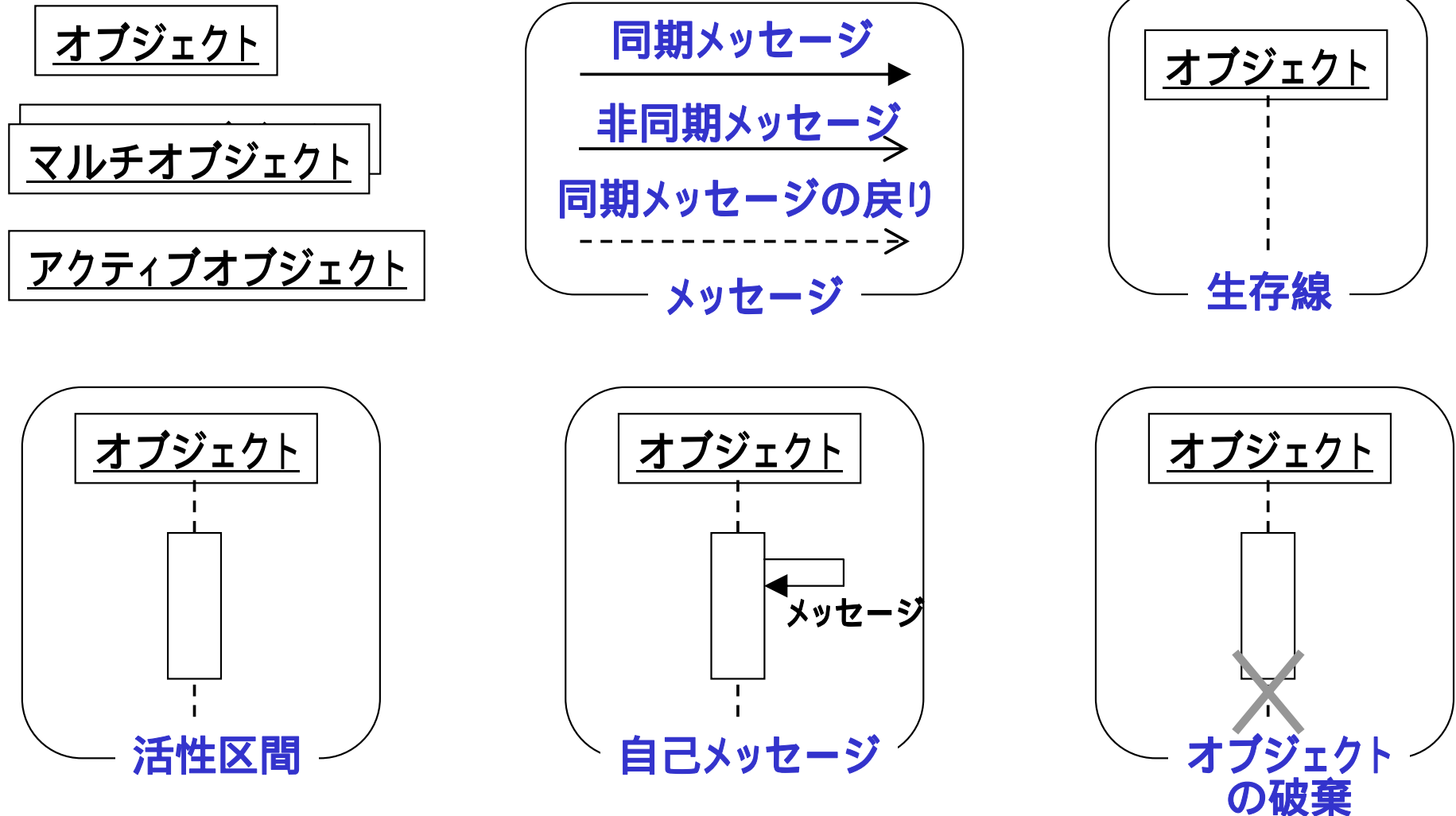
シーケンス図

「相互作用図」のこと
オブジェクト間のメッセージのやり取りを時系列で表す
やりとりする内容・手順・時間を明確にする



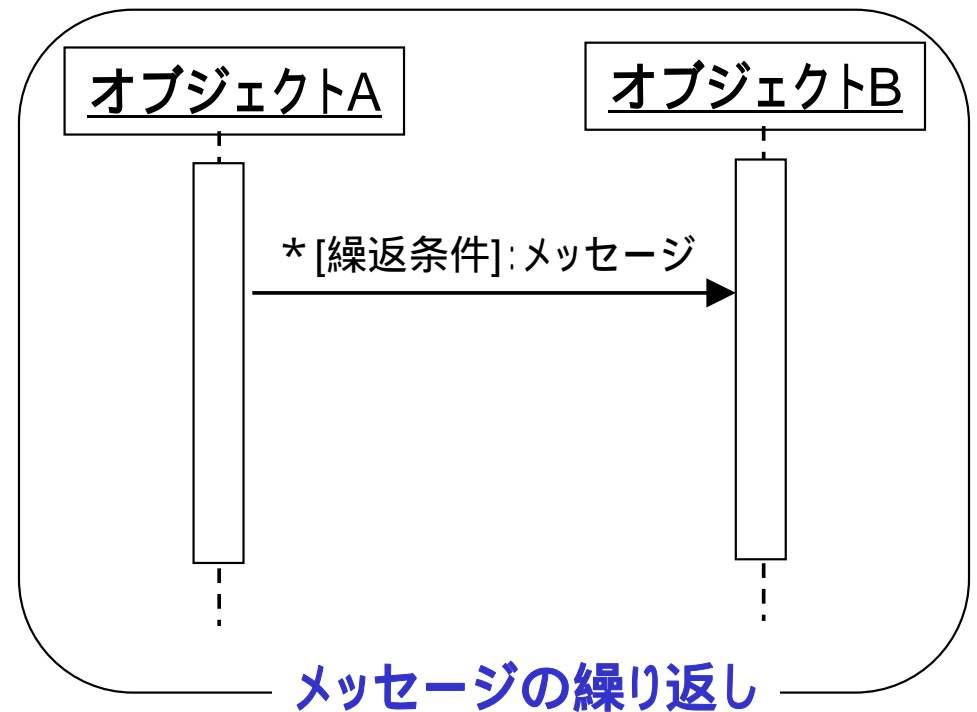
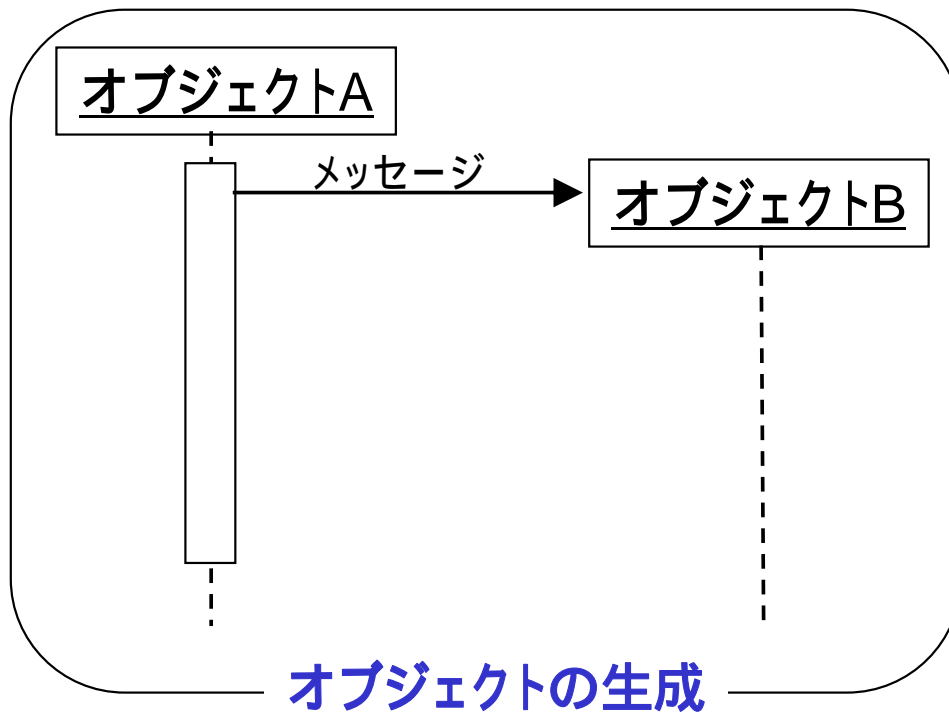
シーケンス図(3)

シーケンス図の要素(1)



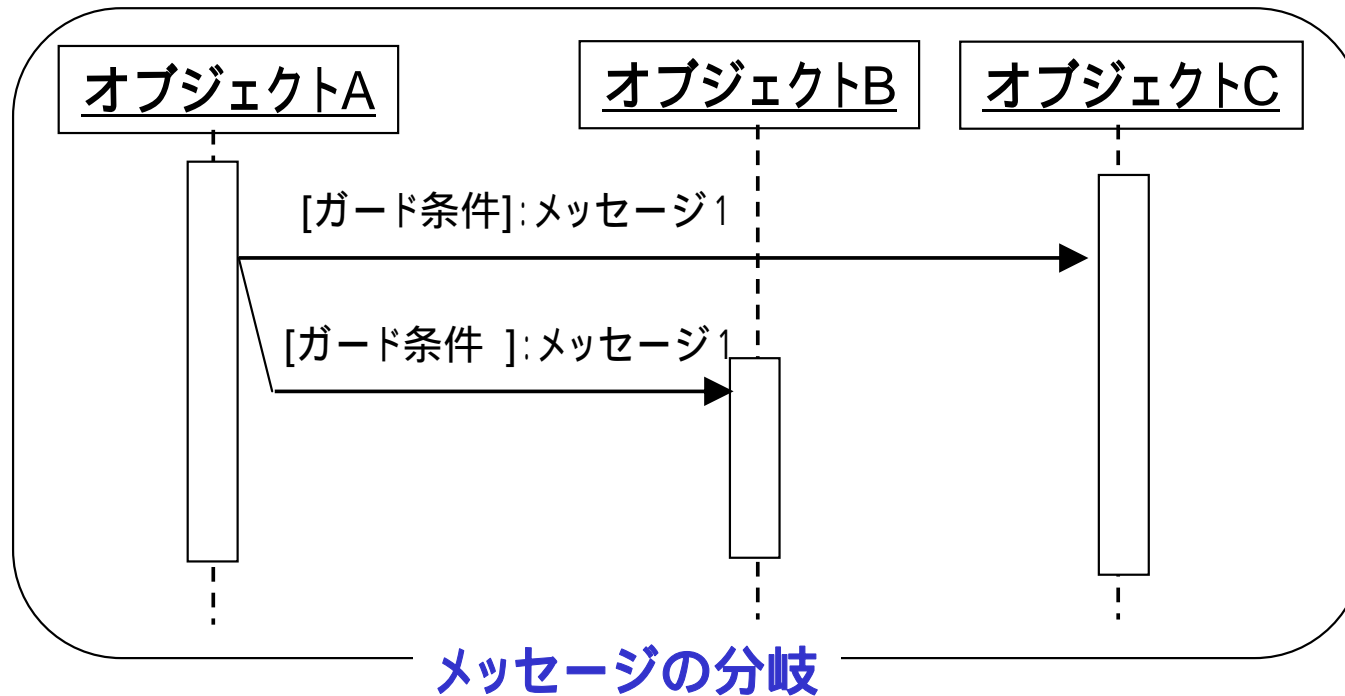
シーケンス図(4)

シーケンス図の要素(2)



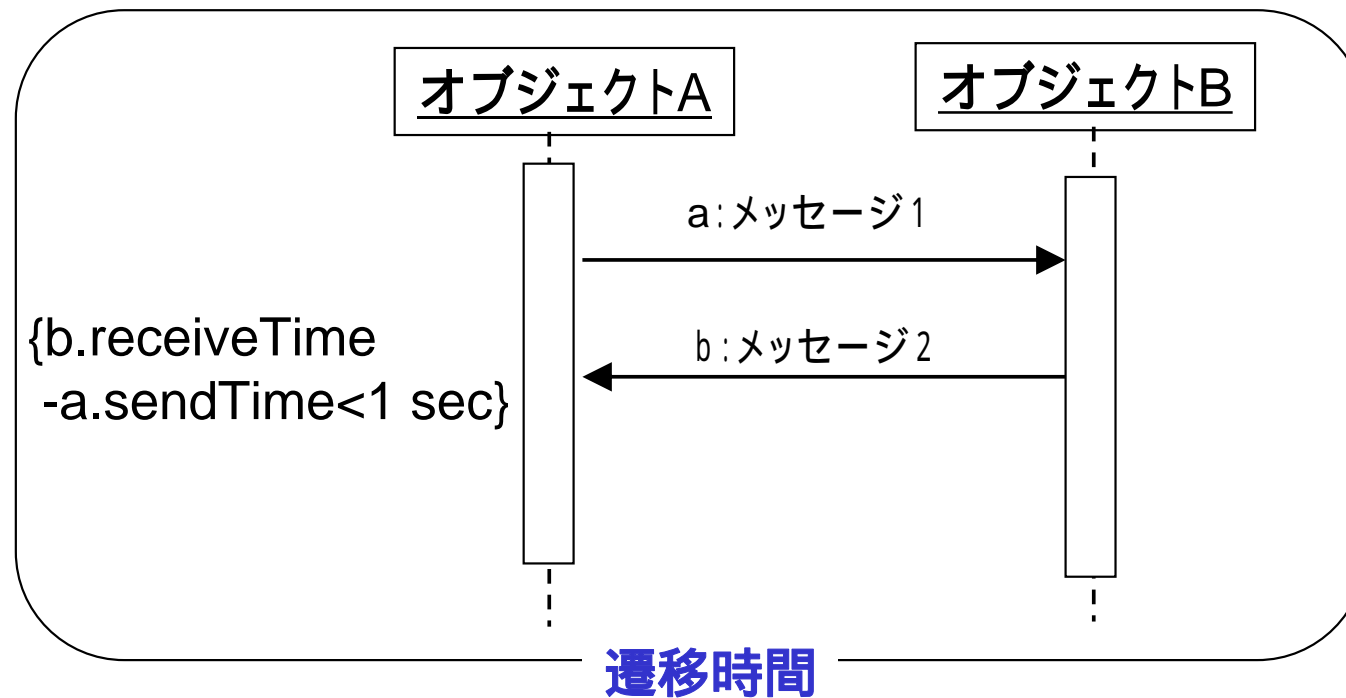
シーケンス図(5)

シーケンス図の要素(3)



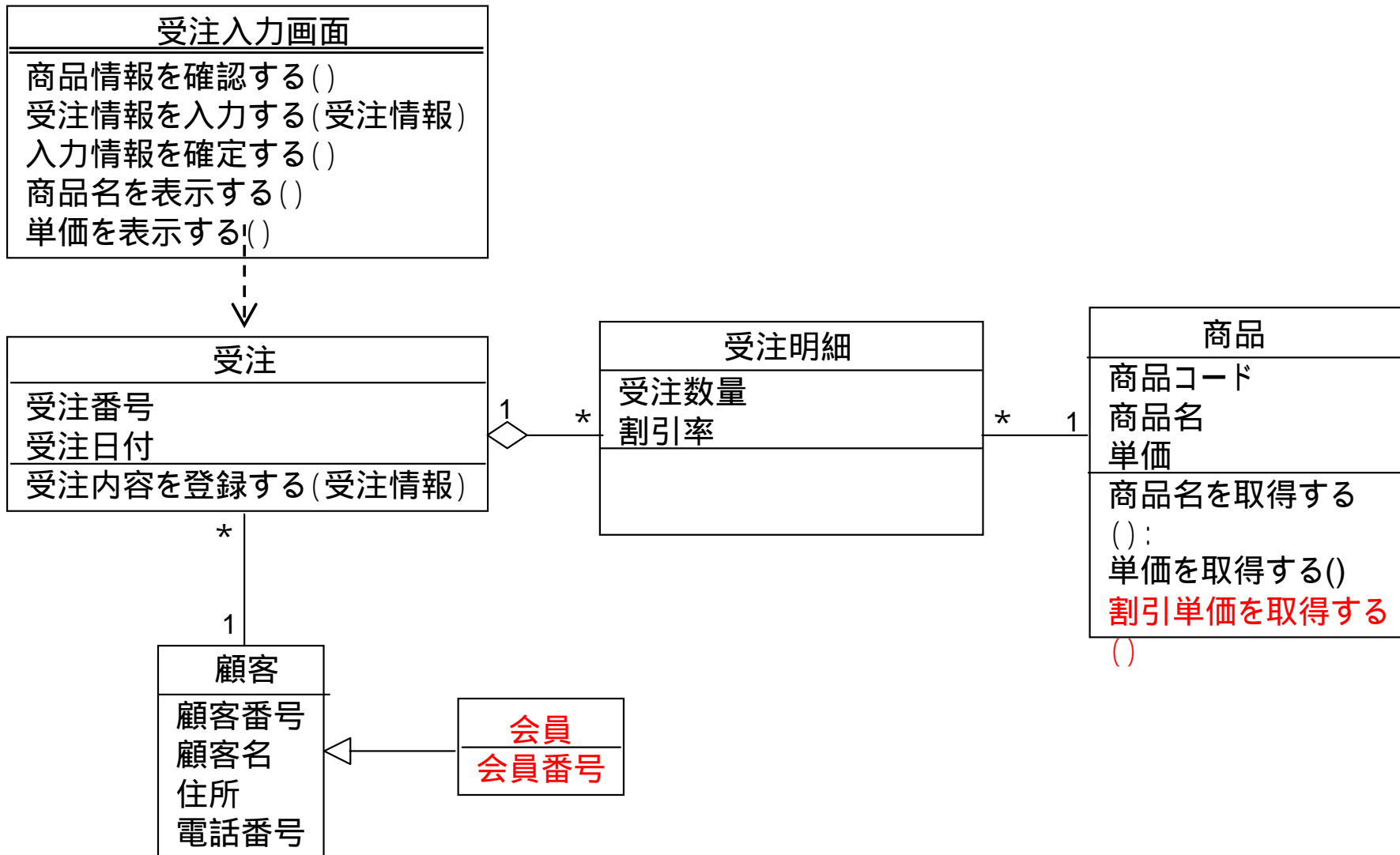
シーケンス図(6)

シーケンス図の要素(4)



クラス図の洗練(2)

顧客クラスを特化して会員クラスを定義



相互作用の追加(2)

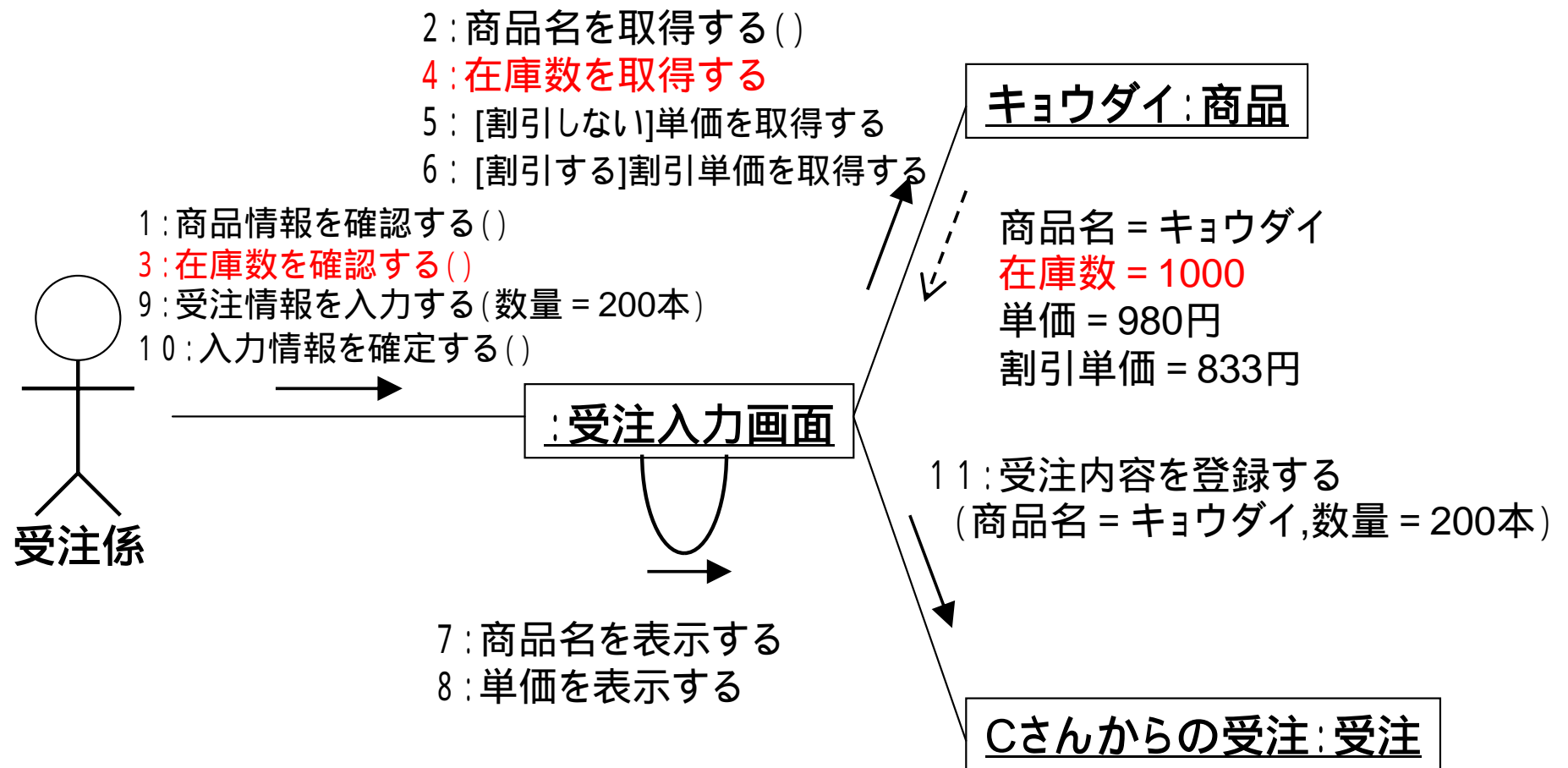
在庫オブジェクトの振る舞いの追加

Bさんから焼酎キョウダイに対して200本の注文があった
各オブジェクト間のやり取りは以下のようなものとする
オブジェクトは受注係、受注入力画面、商品情報、Bさんからの注文とする

1. 入力画面に対して商品情報を確認する
2. 商品情報から商品名を取得する
3. 商品情報から商品名「キョウダイ」が入力画面に出される
4. 在庫情報から在庫数を確認する
5. 在庫情報から在庫数を取得する
6. 「キョウダイ」の在庫1000本と入力画面に返される
7. 単価情報を商品情報から取得する
8. ・Bさんが会員の場合、単価は833円と入力画面に返される
9. ・Bさんが会員でない場合、単価は980円と入力画面に返される
10. 受注入力画面に商品情報を出力する(商品名、単価)
11. 入力画面に対して受注情報(本数)を入力する
12. 入力情報を確定する
13. 入力画面から受注内容をBさんからの注文に登録する

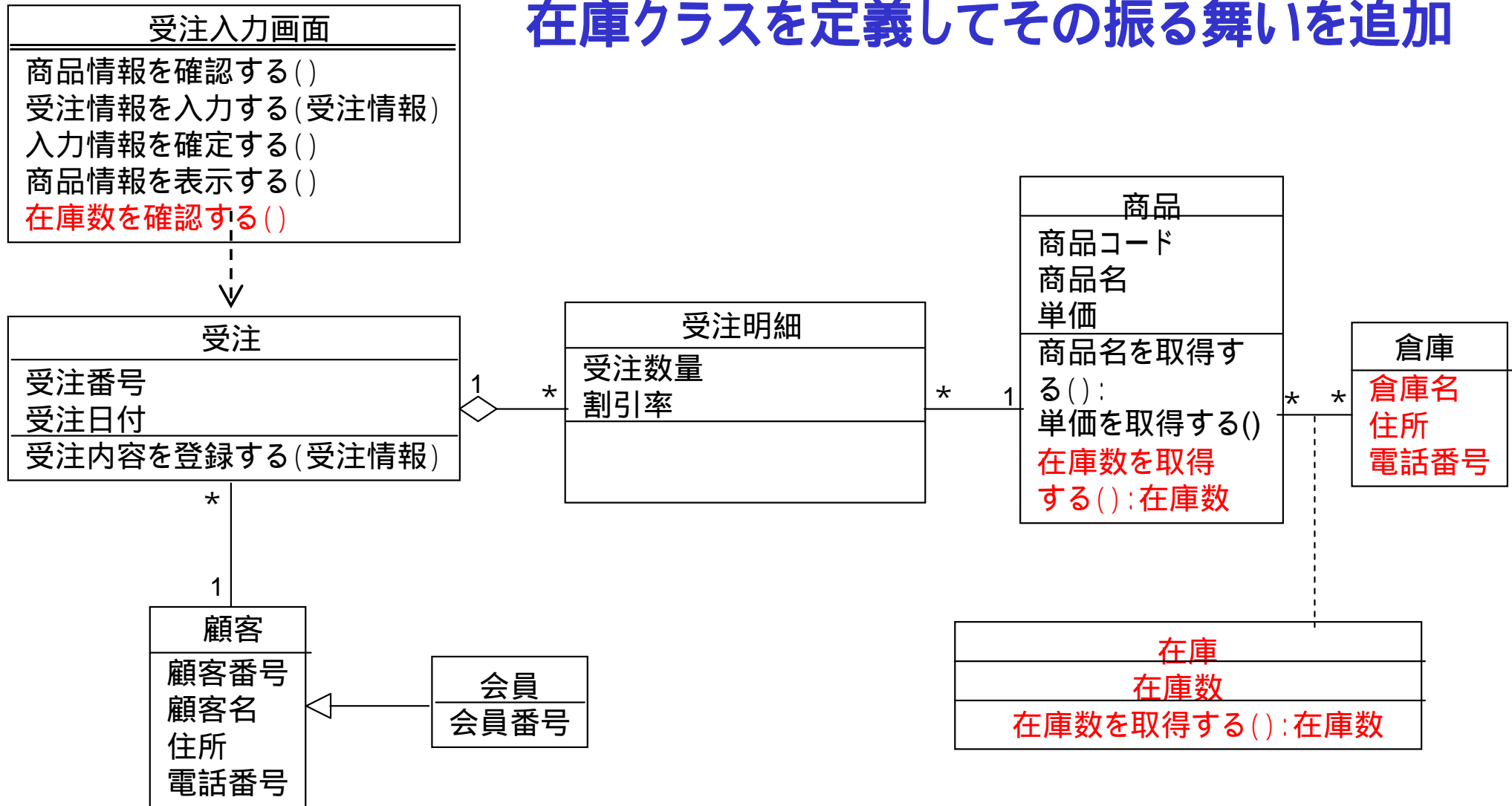
コラボレーション図(7)

在庫オブジェクトの振る舞いの追加



クラス図の洗練(3)

在庫クラスを定義してその振る舞いを追加

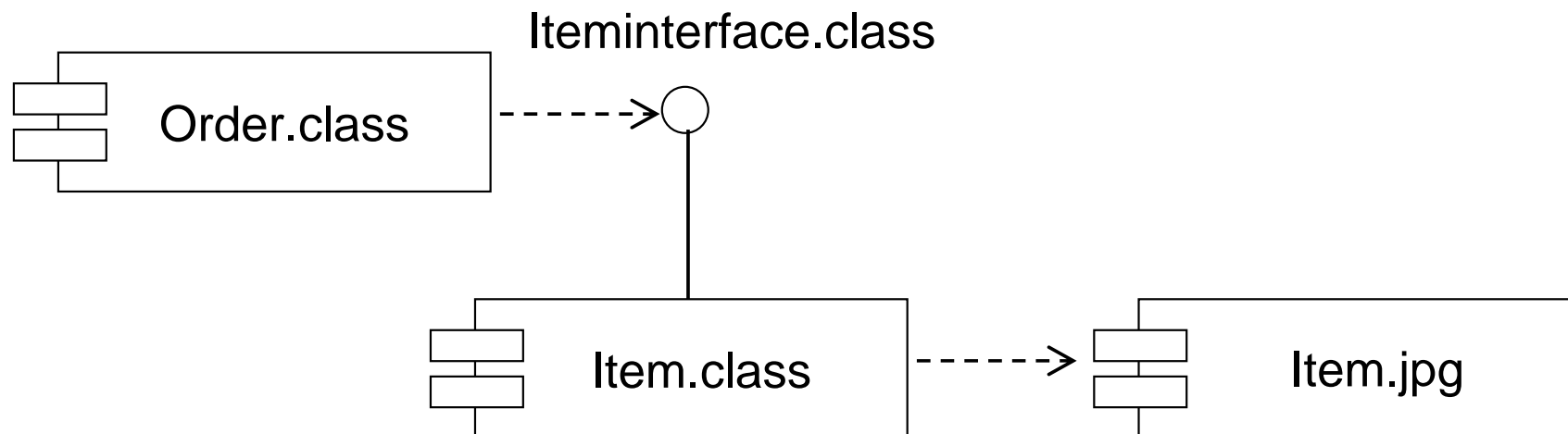


コンポーネント図(1)

コンポーネントの関係

受注を行う (Order.class) は、商品情報を扱う (Item.class) と
中間システムである (Iteminterface.class) を介してデータのやり取りを行う。
Item.classは商品の写真を扱う (Item.jpg) と依存関係にある。
それぞれの関係を考える。

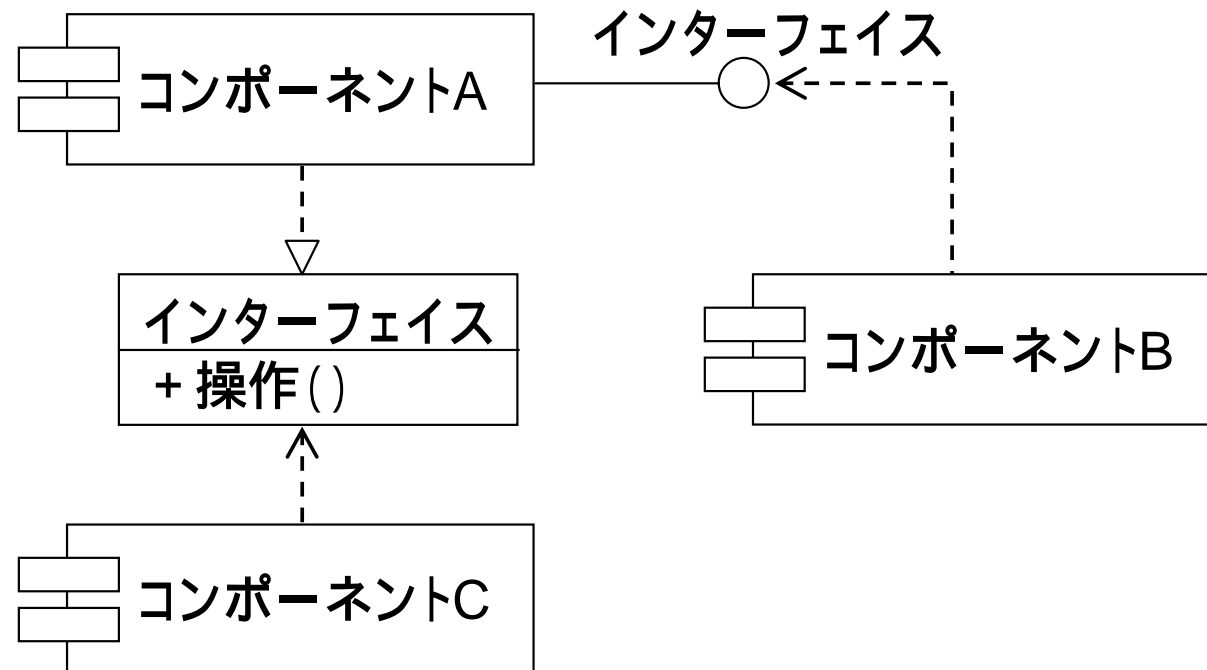
コンポーネント図を使ったモデリング



コンポーネント図(2)

コンポーネント図

ソフトウェア内部の物理的な構成を表す
ソフトウェアのモジュール依存関係やリソースの割当などを
明確化

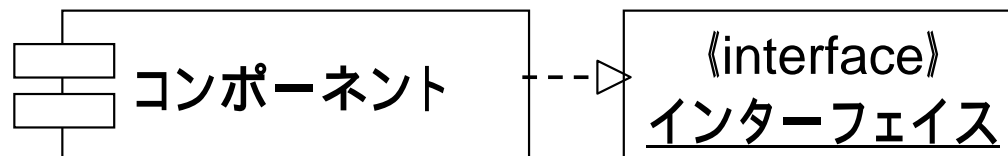
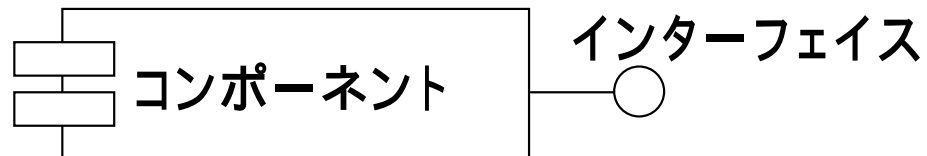


コンポーネント図(3)

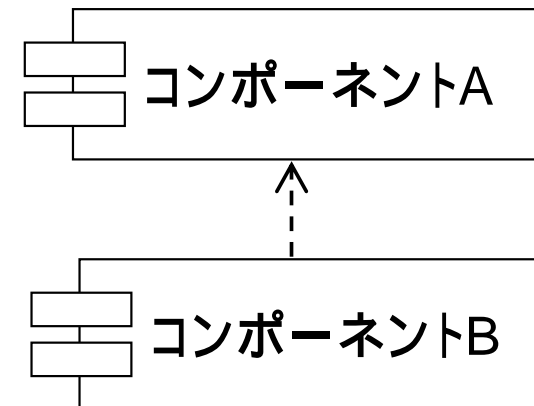
コンポーネント図の要素



コンポーネント



インターフェイス



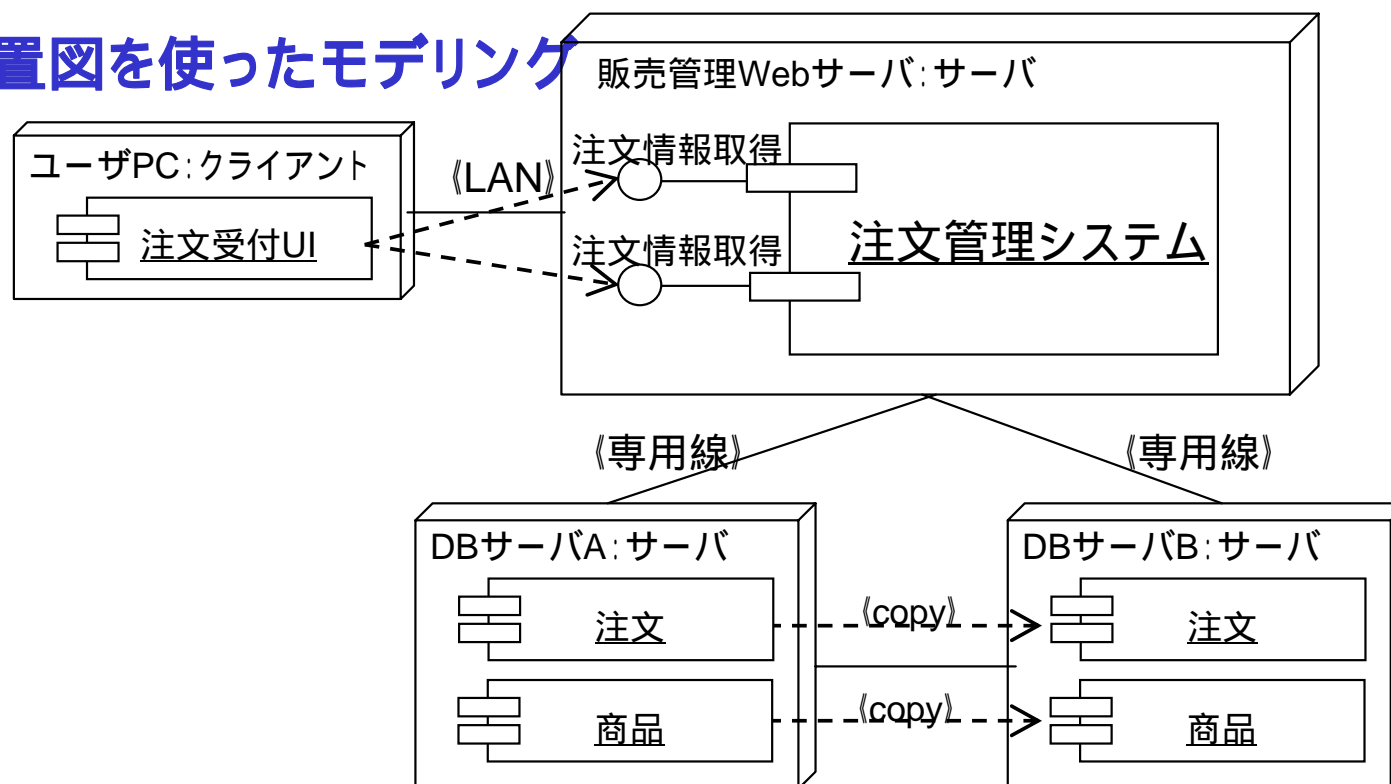
コンポーネント間の依存関係

配置図(1)

システム資源の配置

受注受付UI(ユーザインタフェース)は、ユーザPC(クライアント)にあり、受注管理システムは、販売管理Webサーバにある。両者はLAN回線で接続されている。販売管理Webサーバは、DBサーバAおよびDBサーバBにそれぞれ専用線で接続されている。DBサーバAとDBサーバBは注文と商品の2つデータをそれぞれミラーリング(コピー)している。

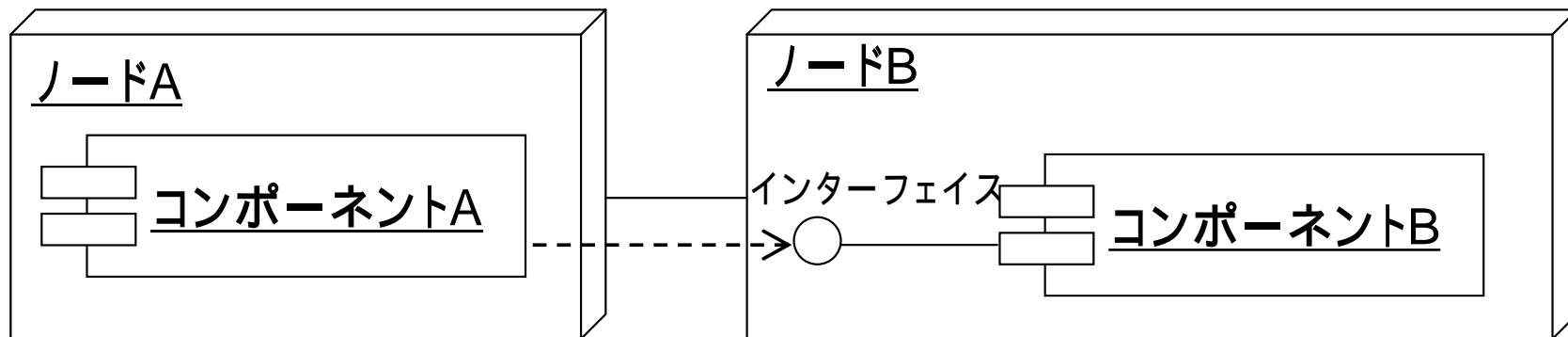
配置図を使ったモデリング



配置図(2)

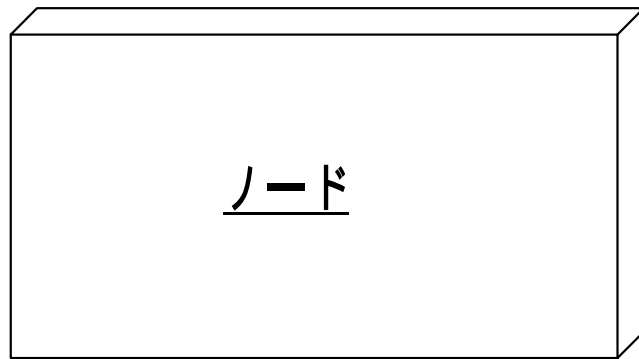
配置図

システムの物理的な構成を表し、実行環境を考えるために
実装段階で作成する
システムのハードウェア構成やソフトウェア構成などを表現



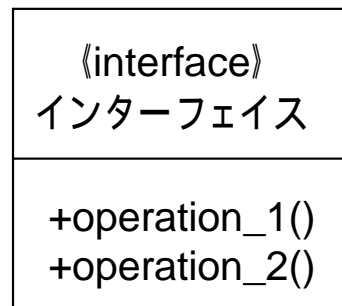
配置図(3)

配置図の要素

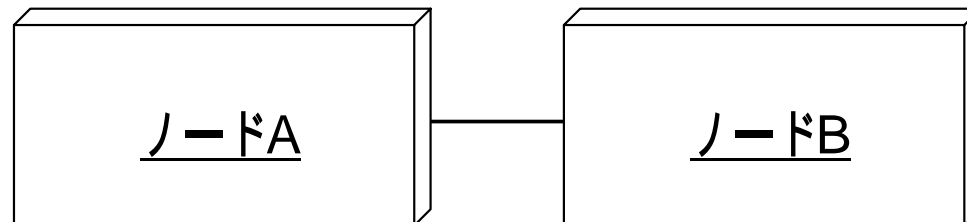


ノード

インターフェイス



インターフェイス



関連

まとめ

- オブジェクト指向によるシステム開発の概要について理解した
 - UMLの概要
 - UMLを使ったシステム開発例

小テスト(氏名:)

- みなさんの研究または関連する研究でオブジェクト指向設計の活用できる場面を考えてください。