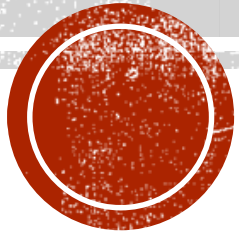
10 ソフトウェア工学
Software Engineering

ソフトウェアモデリング

SOFTWARE MODELING



ソフトウェアモデリングとは？

ソフトウェアの抽象なモデルを，さまざまな視点から簡潔に表現する作業

【表現方法】

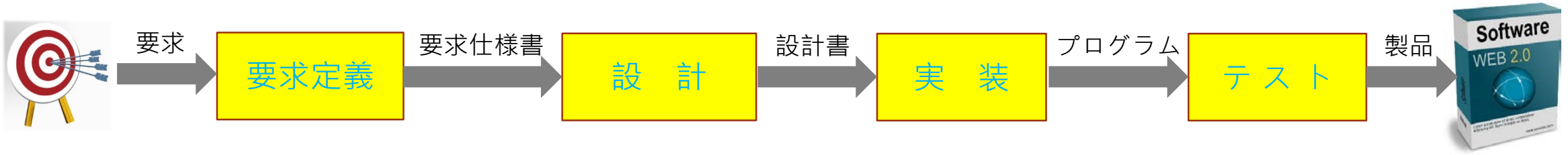
- グラフィカルモデル：所定の記法による図で表現　→ 流れ図，クラス図など
- 数　理　モ　デ　ル：数式や論理式で表現　→ 形式手法 (モデル検査など)

【視　　点】

- 外部からの視点：ソフトウェアが利用される環境のモデル
- 相互作用の視点：システム - 環境・利用者間，システム要素間の相互作用のモデル
- 構 造 の 視 点：システムやデータの静的な構造のモデル
- ふるまいの視点：システムの動的ふるまい，イベントへの反応のモデル

→ 今回の授業では，UML を学ぶ
(さまざまな視点からの記法が国際的に標準化されたグラフィカルモデル)

モデリングをいつおこない，いつ使うのか



【要求定義フェーズ】

- 既存システムのモデリング → 既存システムの明確化 (利点・欠点，再利用性)
- 新 シ ス テ ムのモデリング → 要求の導出と明確化 (関係者との対話・合意)

【設計フェーズ】

- 新システムのモデル → エンジニアへの説明 (設計や文書化にモデルを利用)

【実装フェーズ】

- 新システムのモデル → モデル駆動開発ではモデルから実装の一部を自動生成も
  (model-driven development)

【テストフェーズ】

- 新システムのモデル → モデルに基づいたテストケースの生成

UML (Unified Modeling Language)

- ソフトウェアモデルを表現するためのグラフィカルモデリング言語の１つ
- 13の図式が規定されている
- 1990年代に提案された種々のオブジェクト指向モデリング図式を統一したもの
- 1997年に初版 UML 1.0 が公開
- 最新版は2015年に公開されたUML 2.5

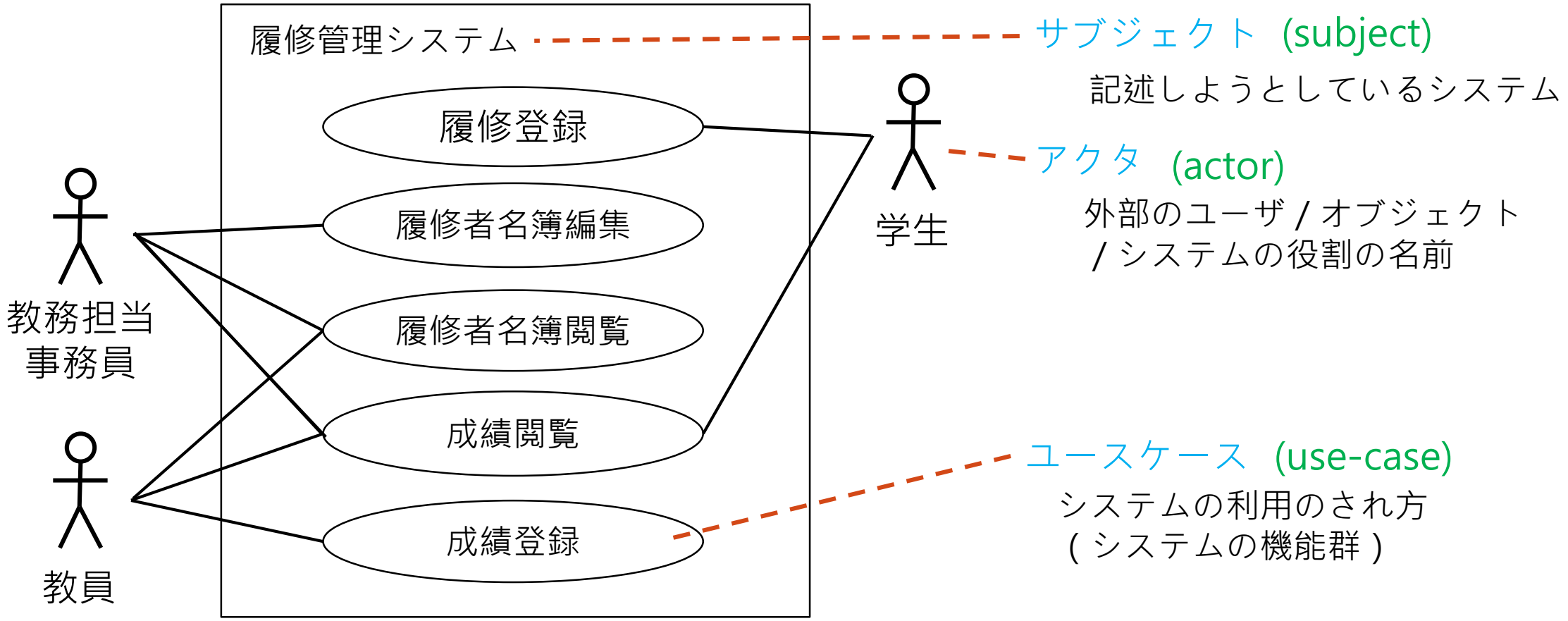
UMLの主な図式

■13の図式のうち，今回はつぎの5つを学ぶ

- ユースケース図　(use-case diagram)　システムの機能群と利用者の関係
- クラス図　(class diagram)　クラスの定義とクラス間の関係
- アクティビティ図 (activity diagram)　システムがおこなう処理の流れ
- シーケンス図　(sequence diagram)　オブジェクト間のメッセージ交換の順序
- 状態機械図　(state machine diagram)　イベントに伴うオブジェクトの状態遷移

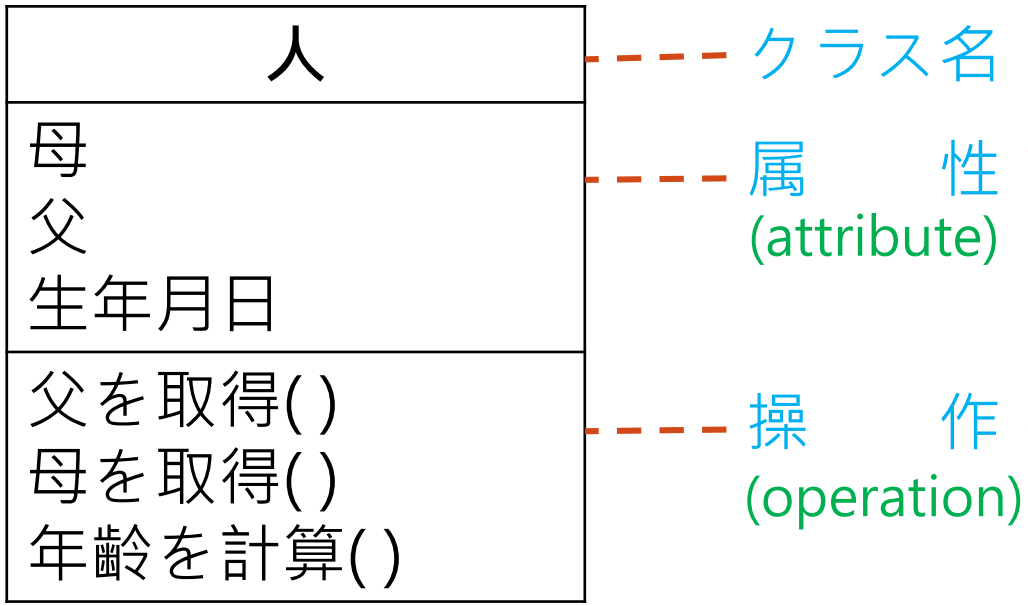
ユースケース図　システムの機能群と利用者の関係を図示

【履修管理システムの例】



クラス図 (1/9)　クラスの定義とクラス間の関係を図示

【クラスの定義】(簡易版：要求定義段階)



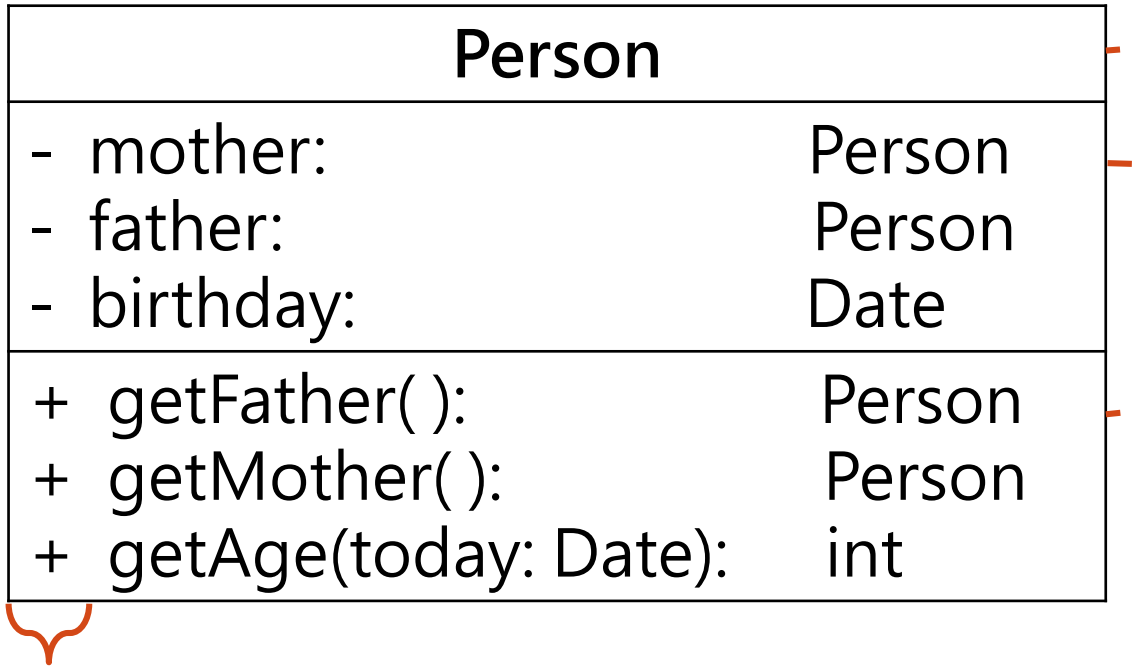
※属性と操作は場合によっては省略可

Javaでは
　フィールド(field)
C++では
　メンバ変数(member variable)
抽象化され直接アクセスできないときは
　プロパティ(property)
と呼ばれる

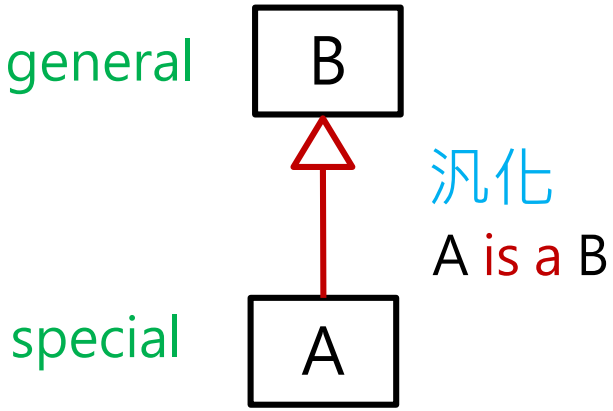
Javaでは
　メソッド(method)
C++では
　メンバ関数(member function)
と呼ばれる

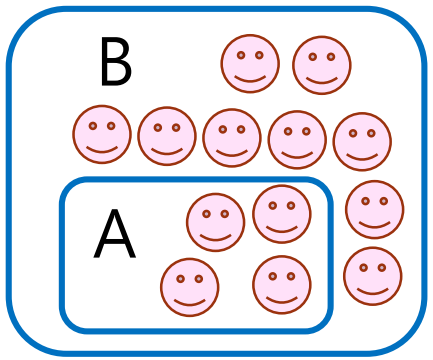
クラス図 (2/9)

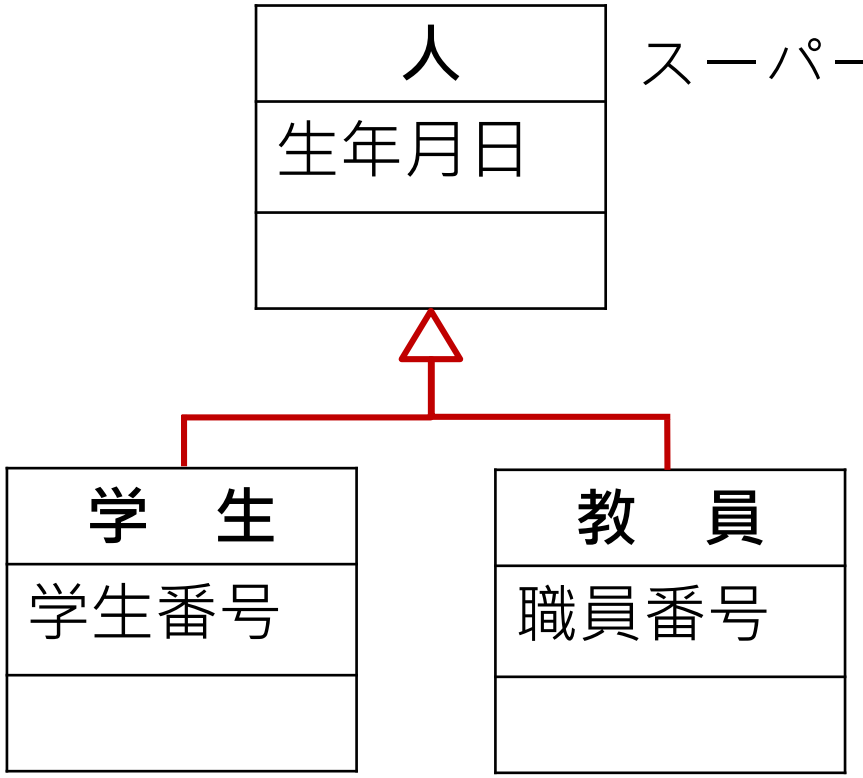
【クラスの定義】（詳細版：プログラム設計段階）



クラス図 (3/9)　【クラス間の関係：汎　化】
(generalization)







スーパークラス

サブクラス

【継承】(inheritance)

- スーパークラスの属性と操作は引き継ぐ
- 新しい属性と操作を追加可能

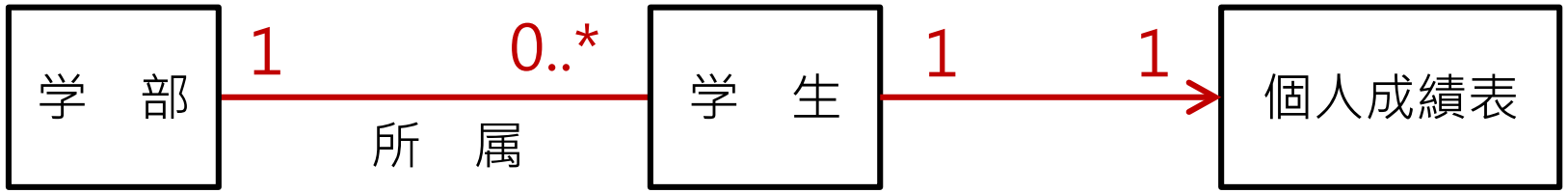
クラス図 (4/9)　　【クラス間の関係：関　連 (association)】

A is associated with B

多重度 (multiplicity)
クラスAのインスタンスを１個固定したとき
クラスBのインスタンスが n 個対応

n	必ず n
n..m	n 以上 m 以下
0..*　または　*	0 以上
1..*	1 以上

クラスAのインスタンスからクラスBのインスタンスにアクセスできる



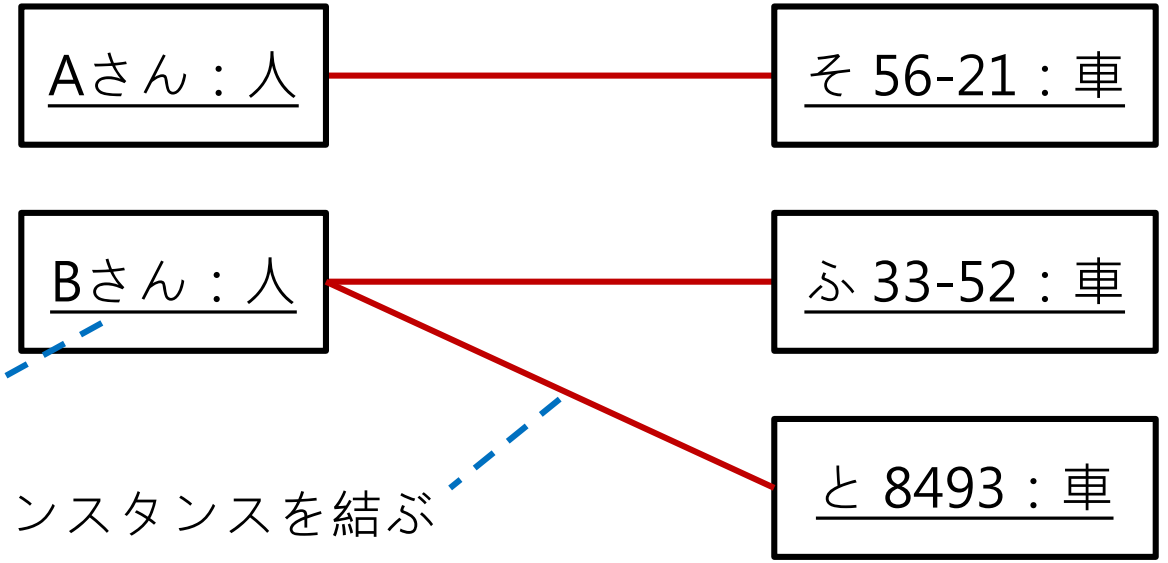
クラス図 (5/9)

【オブジェクト図】
(object diagram)

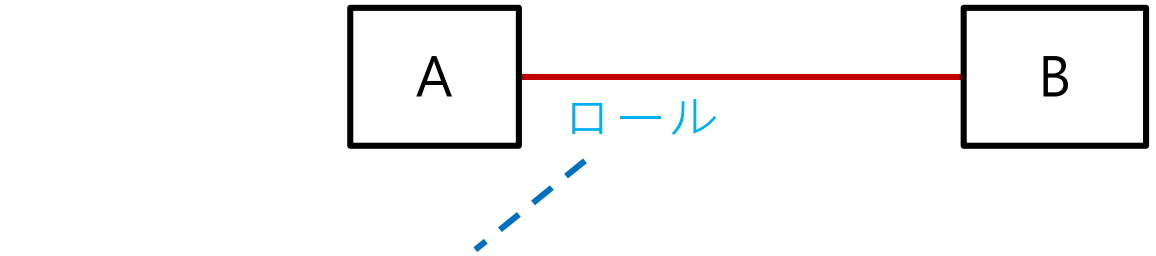
クラス図

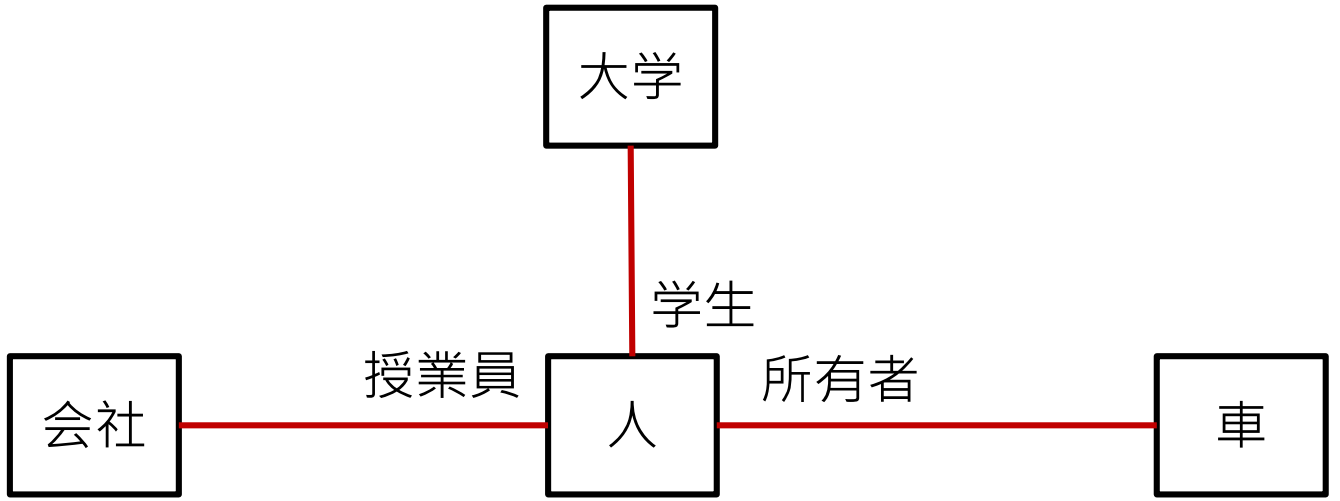
オブジェクト図

クラス図の表す抽象構造の１つの具体例の表示



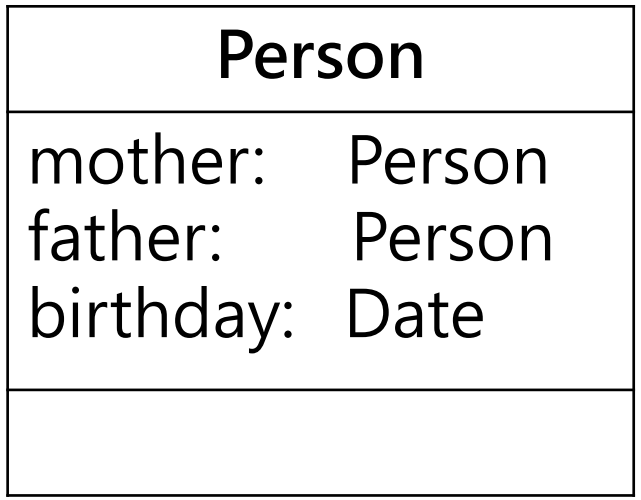
クラス図 (6/9)　【ロール】
(role)

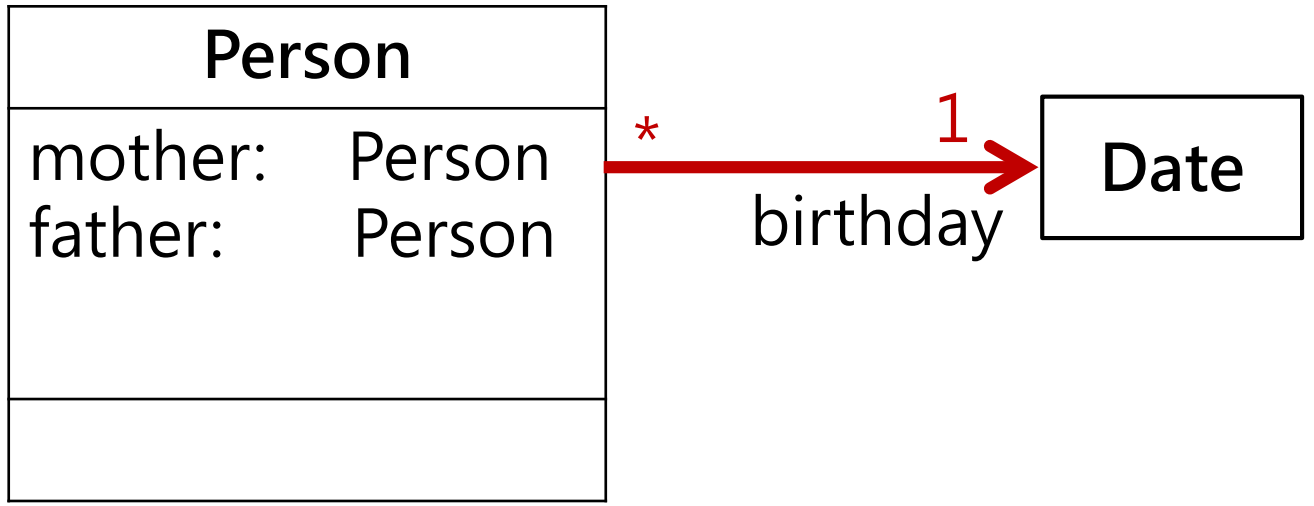


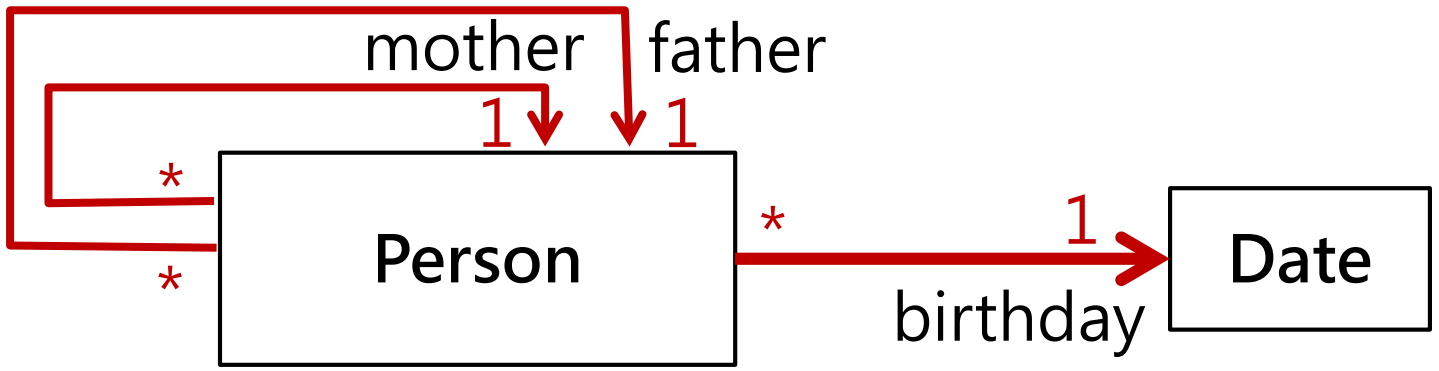


クラス図 (7/9)　【属性 / 関連】

属性，関連：どちらで表現してもよい / 混在させてもよい





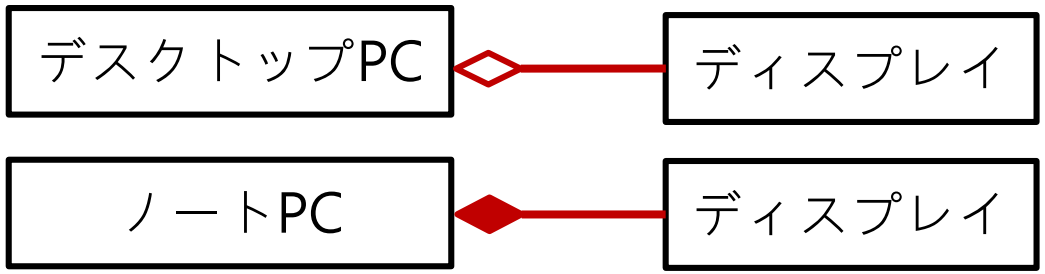


クラス図 (8/9)　【集約 / コンポジション】
(aggregation / composition)

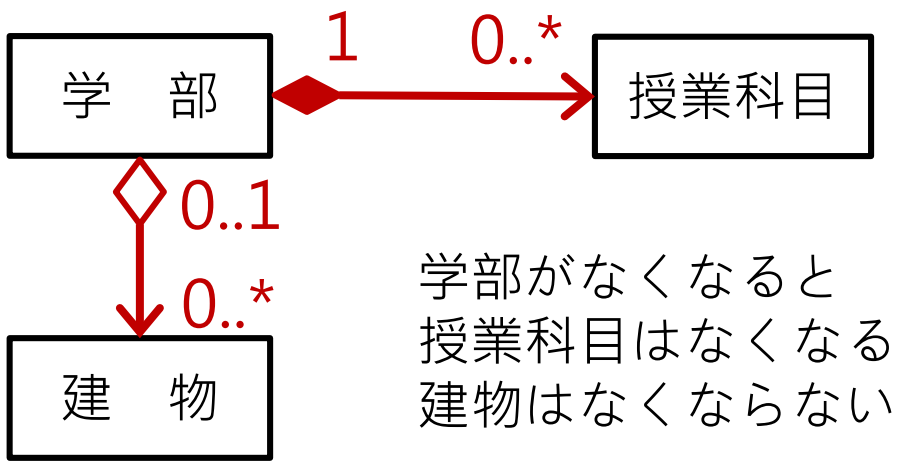
A has a B

全体 - 部分 の関係 (whole-part)
クラスBはクラスAの一部

強い集約関係
クラスAのインスタンスが削除されると
クラスBのインスタンスも削除



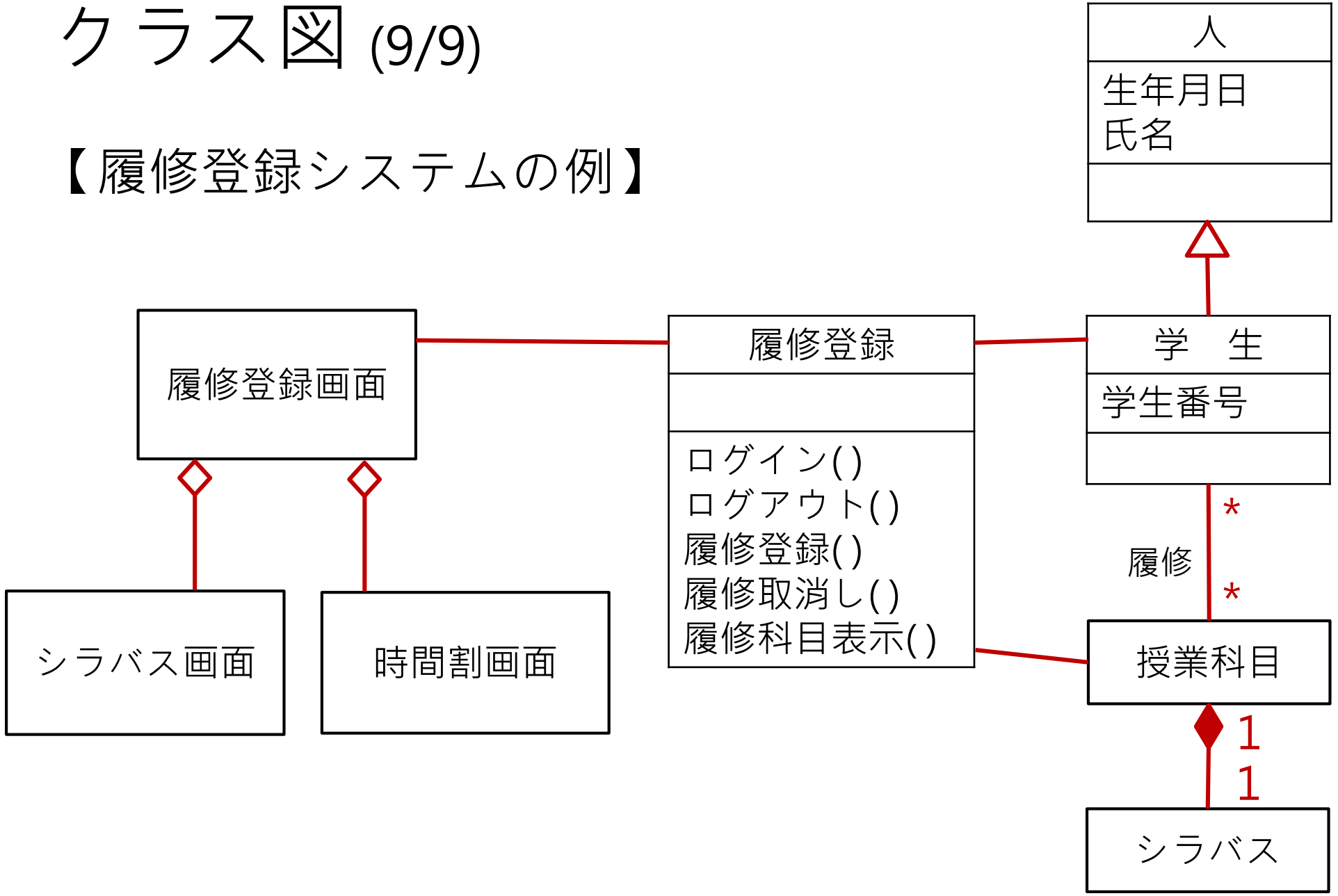
PCを廃棄するとき，ディスプレイも廃棄？
　デスクトップPC:　NO
　ノートPC:　　YES



学部がなくなると
授業科目はなくなるが
建物はなくならない

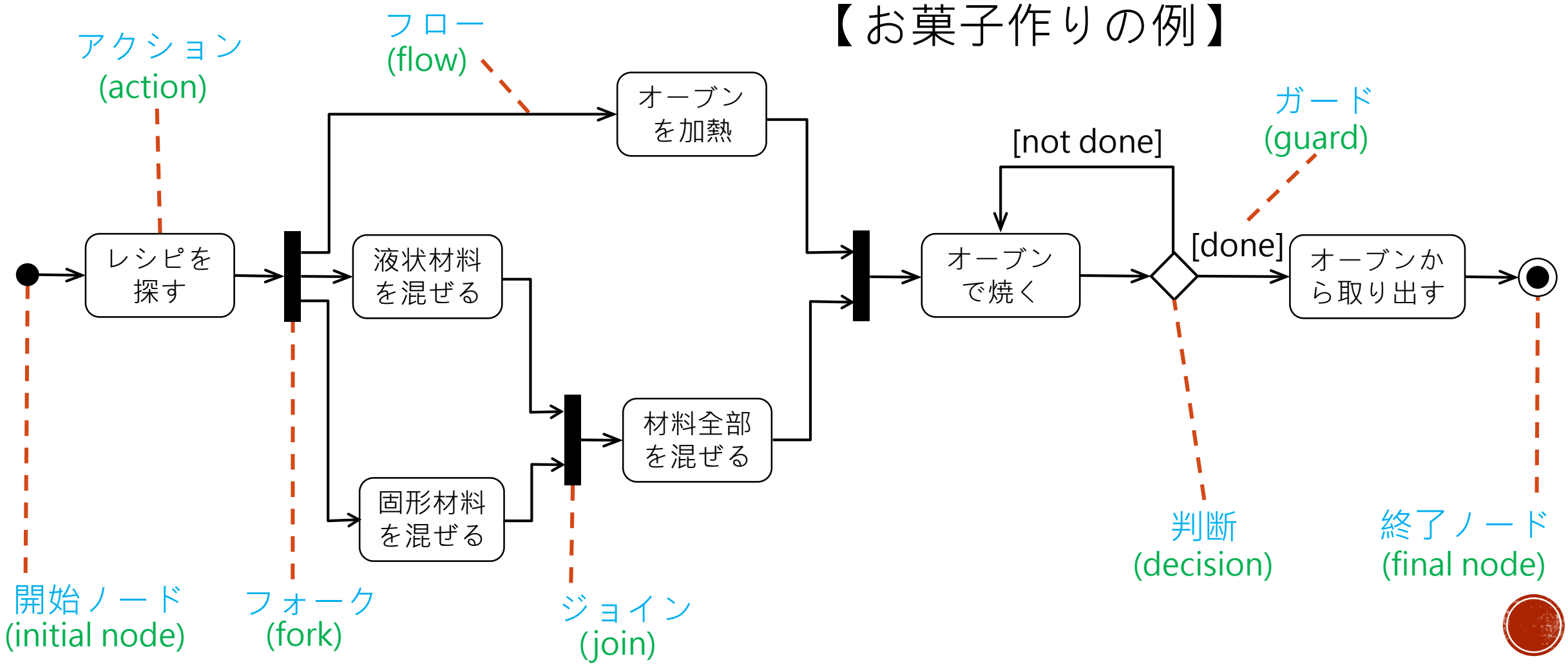
クラス図 (9/9)

【履修登録システムの例】

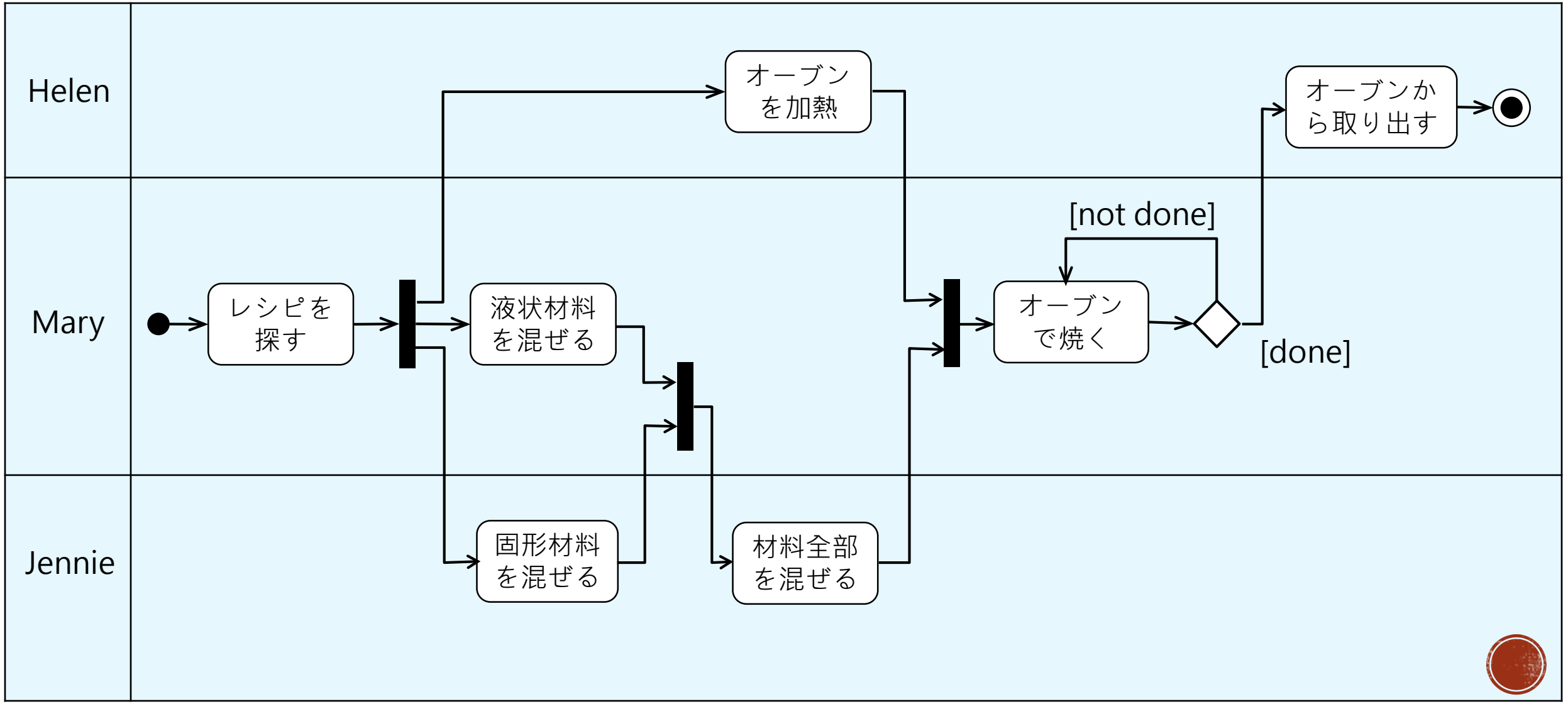


アクティビティ図 (1/2)　システムがおこなう処理の流れを図示

アクティビティ (activity)：システムの振る舞いをアクションのフローとして記述したもの

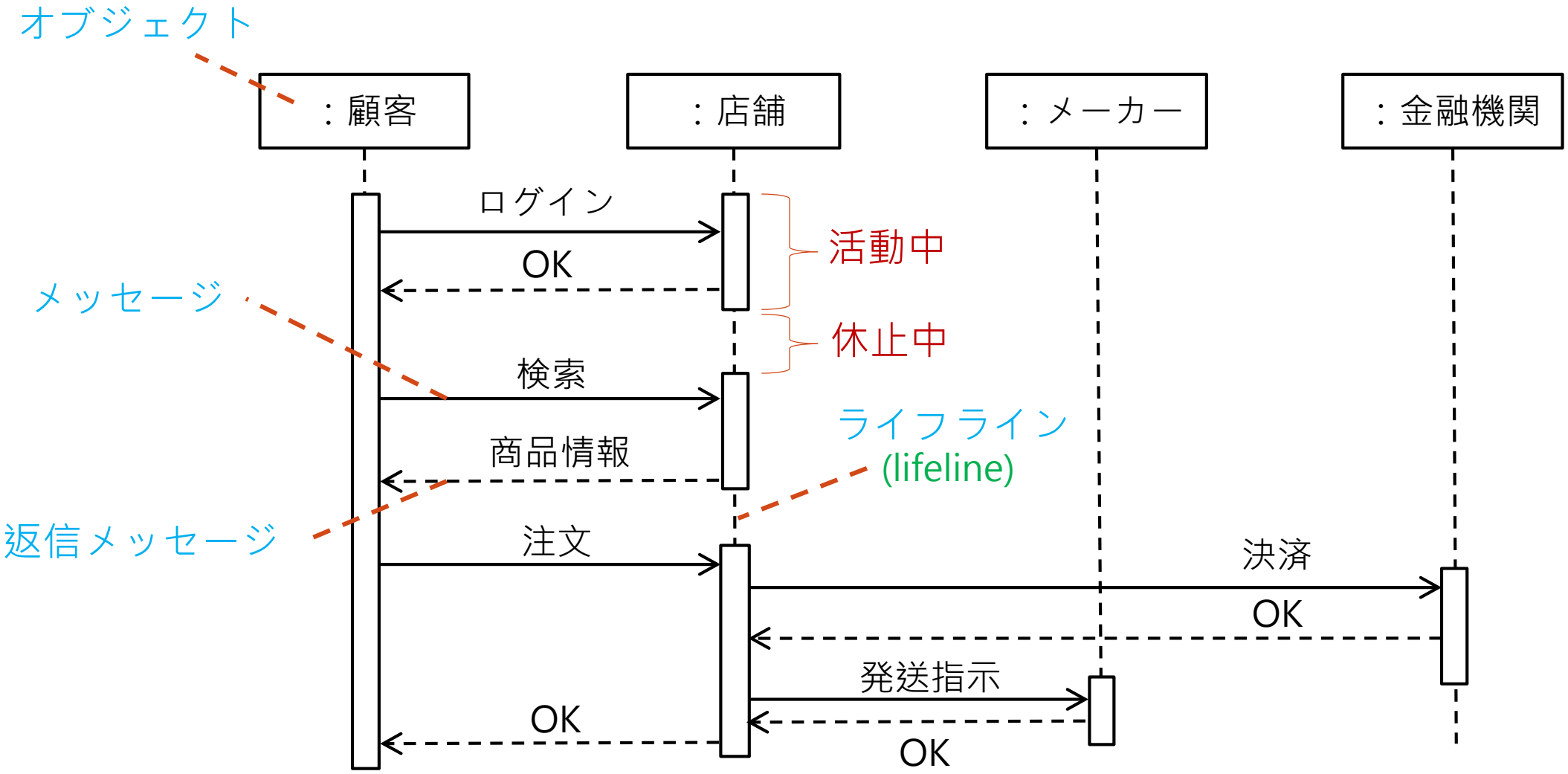


アクティビティ図 (2/2)　スイムレーンによる実行主体の明示
(swimlane)



シーケンス図　オブジェクト間のメッセージ交換の順序を例示

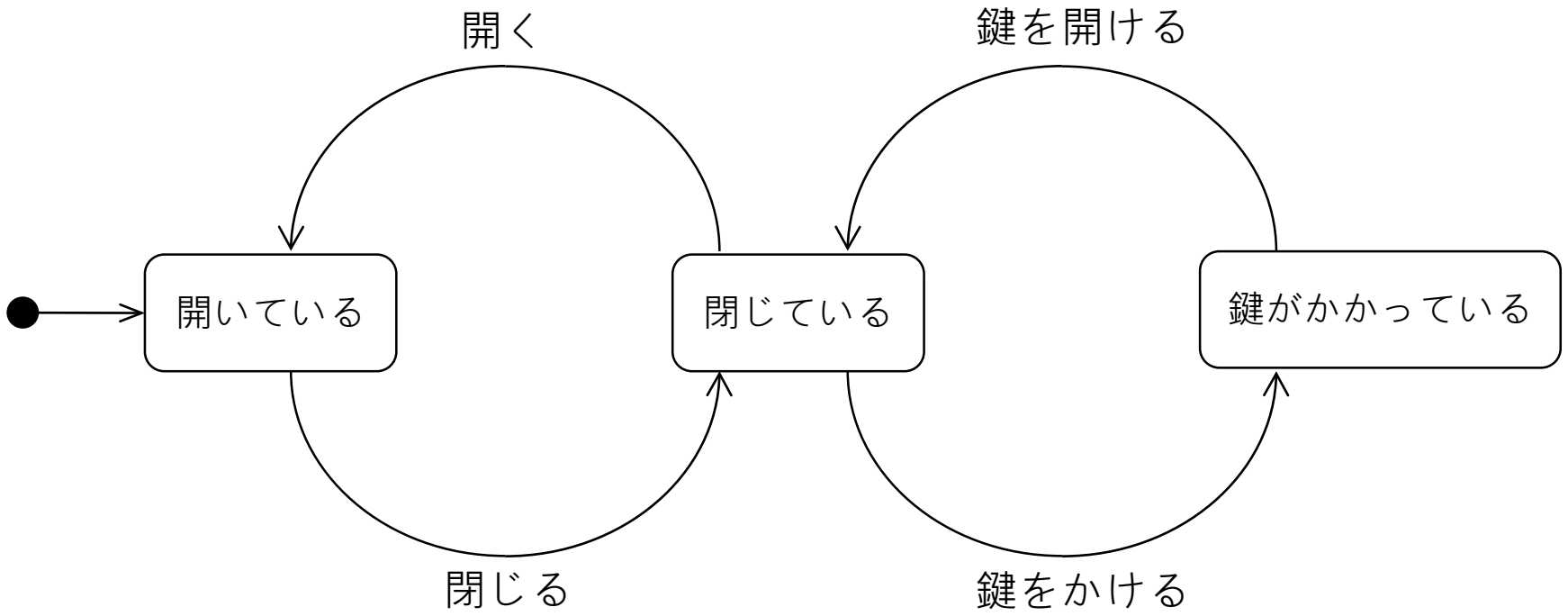
【オンラインショッピングの例】



状態機械図

イベントに伴うオブジェクトの状態遷移を図示

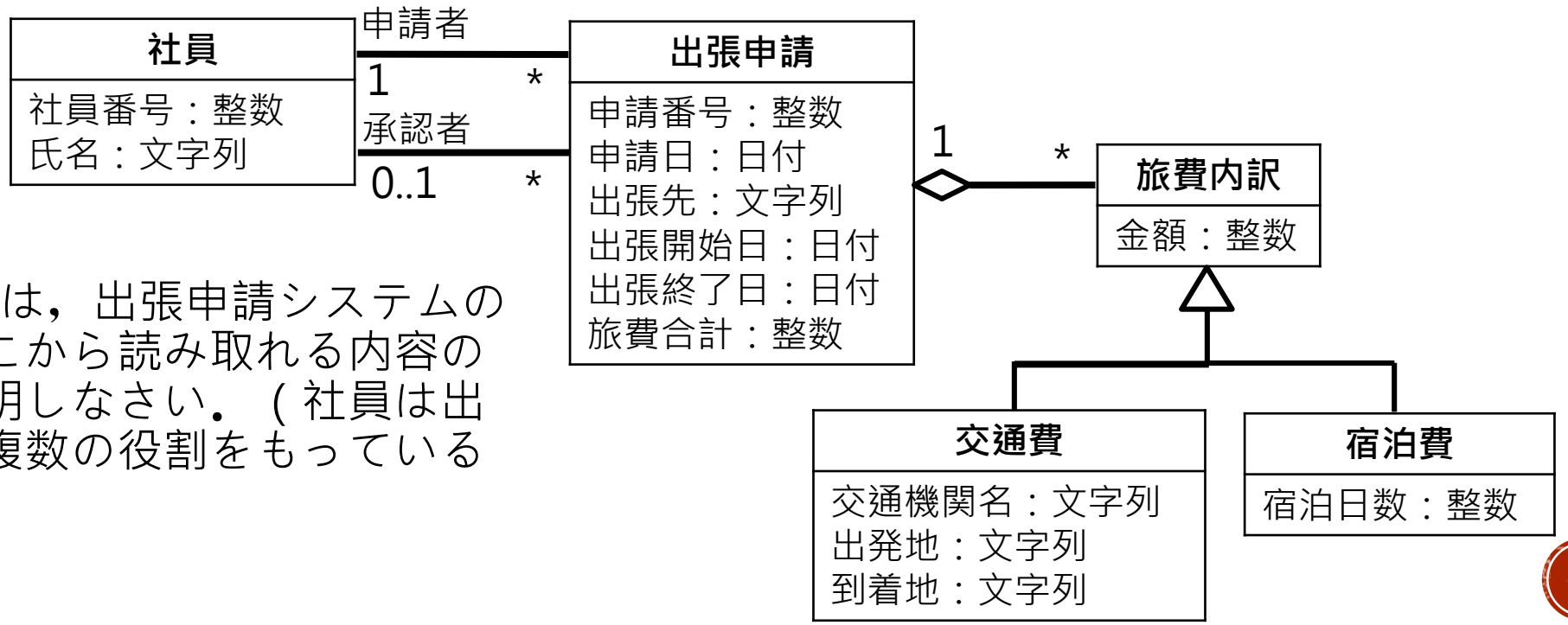
【ドアの例】



演習問題 10

(1) つぎのシステムのユースケース図を作成しなさい．

チケット予約システムは，チケットのオンライン予約を管理するものである．利用者は，チケットの予約，支払い，予約状況の確認，チケットの印刷を行うことができる．管理者は，販売可能なチケットの総数の設定，予約状況一覧表の表示，チケットの印刷を行うことができる．



(2) 右のクラス図は，出張申請システムの一部である．ここから読み取れる内容の概略を簡単に説明しなさい．（社員は出張申請に対して複数の役割をもっていることに注意）