オブジェクト指向論(Q)

オブジェクト指向概論(B1)
オブジェクト指向(K1)

講義資料#5
2023/5/8

來村 徳信

第1回講義
復習

モデルの分類の観点

- 「 時間的」観点からの分類：
  - (1) 「静的」モデル（UML：「構造図」）
    - 対象世界になにが存在し，どのような関係があるかなどを表す．
    - 時間を気にしない．存在しうるオブジェクトや関係を全部，記述する
    - オブジェクトの「性質」を表す
  - (2) 「動的」モデル（UML：「振る舞い図」）
    - 時間が流れている
    - 対象世界やソフトウェアがどのように動くか（振る舞い）を表す
    - オブジェクトの「動作／処理／相互作用」を表す
- 「 内部 ／ 外部 」の観点からの分類：
  - (1) 「外部的」観点からのモデル
    - 情報システムを外部（ユーザの観点）から見たモデル．機能を表す．
    - 情報システムが内部的にどう動くかから独立
    - 分析フェイズや設計フェイズで用いられる
  - (2) 「内部的」観点からのモデル
    - 情報システムが内部的にどう動くかを表す
    - 設計フェイズの後半の詳細設計で用いられる

これまでの講義

これからの講義

第1回講義
復習

UMLのモデル図の種類

- 13種類ある
- 構造図（＝ 静的モデル）
  - クラス図，オブジェクト図，パッケージ図，コンポーネント図，コンポジット構造図，配置図
- 振る舞い図（＝ 動的モデル）
  - 「相互作用図」
    - シーケンス図，コミュニケーション図，相互作用図，タイミング図
  - ユースケース図，アクティビティ図，ステートマシン図

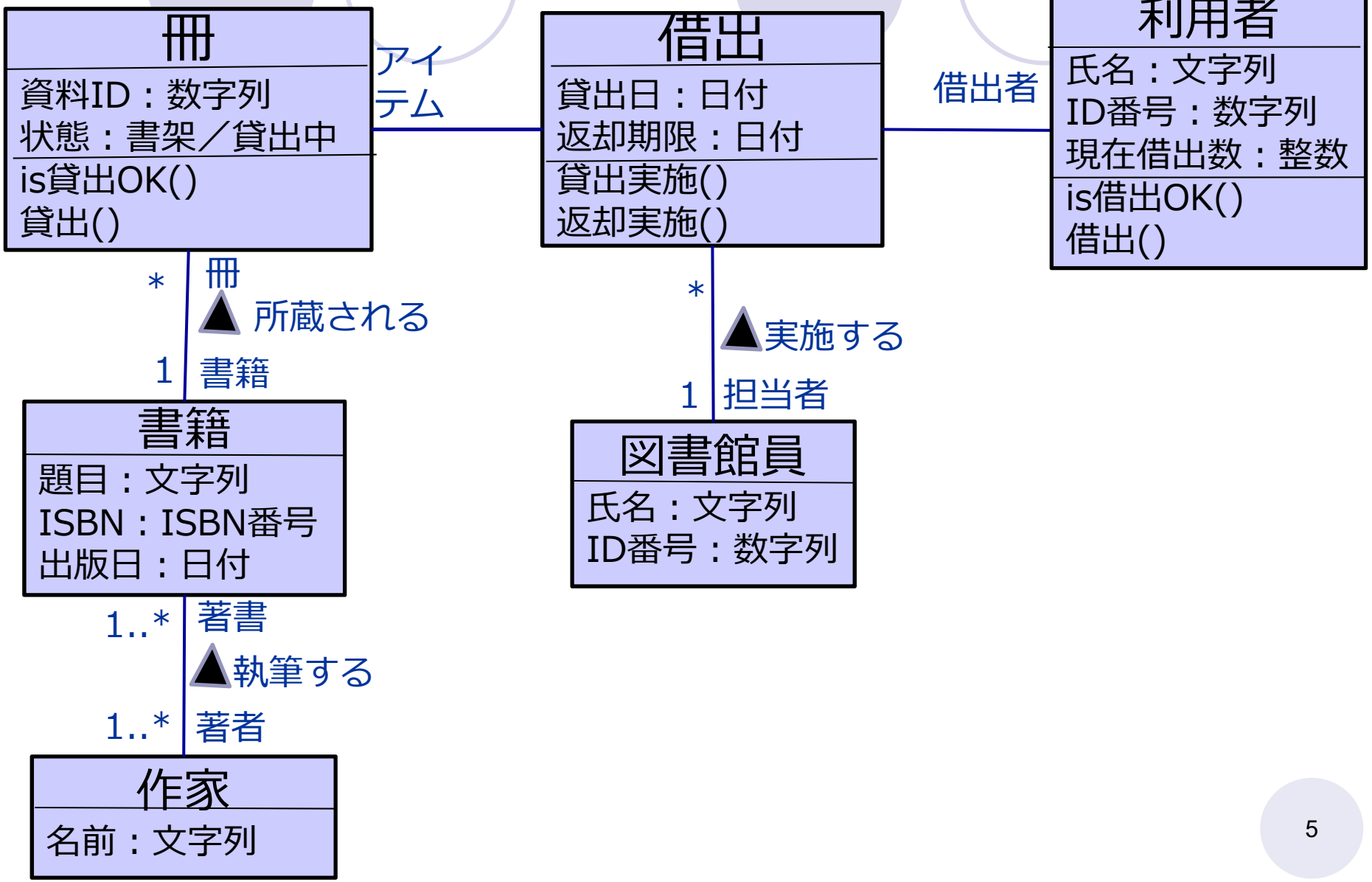
- 本講義では下線のモデル図（のみ）を扱う．

テーマと流れ

- 外部的観点からの動的モデル
  - 外部的観点：システムを外部から捉える
  - 動的：時間を明示的に扱う
  - ユースケース図
    - 機能的要求，使用シナリオ
- 内部的観点からの動的モデル
  - アクティビティ図
    - 内部的観点から「処理（オブジェクトの動作）の流れ」を表す
    - 外部的観点からの「ワークフロー」の表現にも使われる
  - 次 シーケンス図
    - アクターやオブジェクト間の「対話」を表す
    - メッセージパッシング

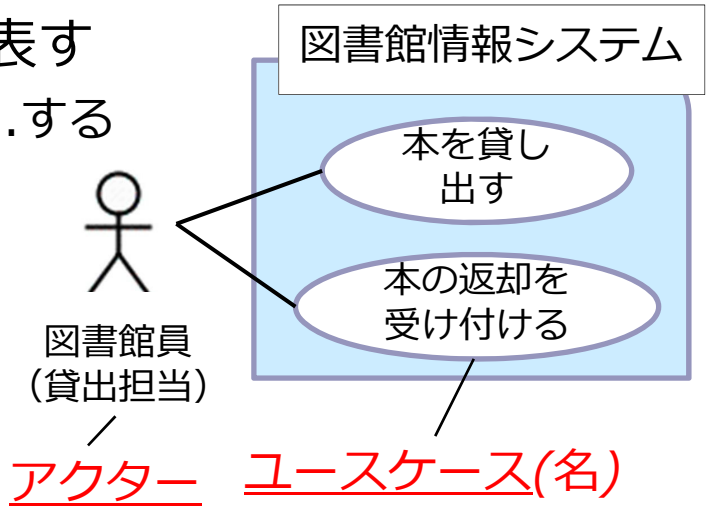
例題：図書館情報システム

- クラス図



ユースケースとは

- ユーザがシステムを「使用」する「シナリオ」
  - 「アクター」　：システムの利用者（人間・組織・他のシステム）の役割
  - 「ユースケース」（ユースケース「名」）：
    - 基本的に「対象物」＋「 動詞」で表す
      - アクターが主語．アクターが～を．..する
    - ユースケース図に書かれる
  - ユースケースの「内容」　：
    - ユーザがシステムを使うときの１つの「シナリオ」を表す
      - アクターとの「やりとり」を表す
      - 「本を貸し出す」ユースケースのシナリオの例：
        (1) 図書館員がシステムに利用者と本のIDを与える
        (2) システムが貸出OKかどうかを表示する
        (3) システムが貸出を記録する

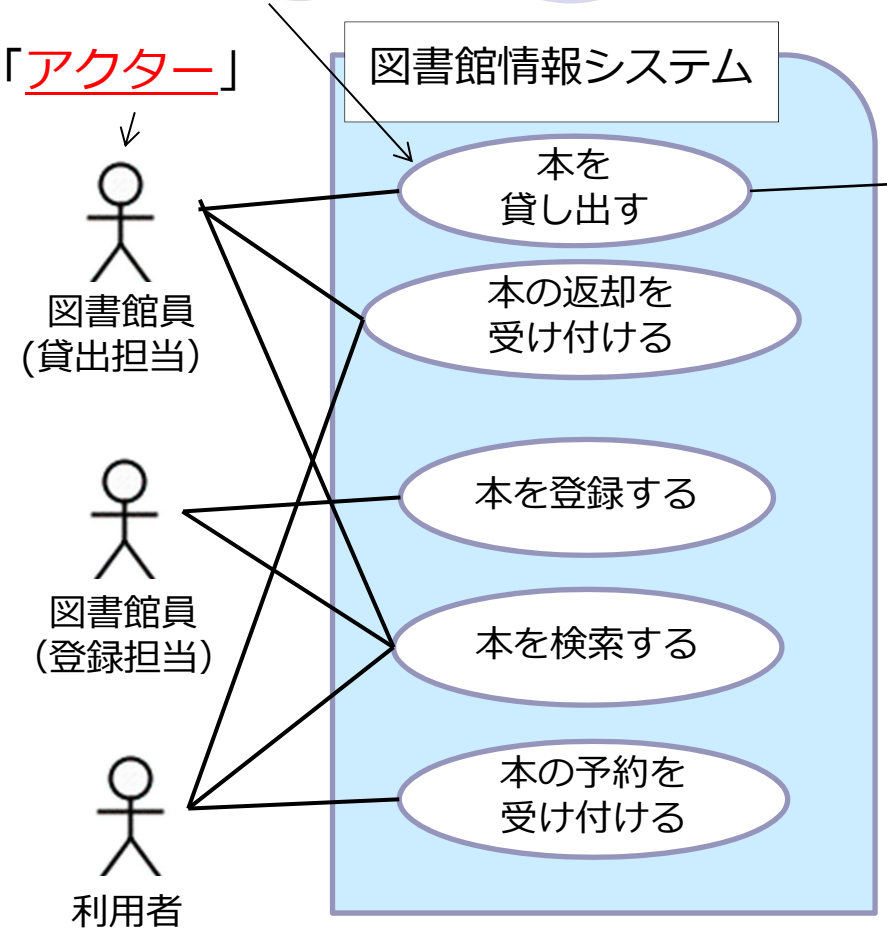


ユースケース図とユースケースの内容

「ユースケース(名)」

「アクター」

各ユースケースの「内容」

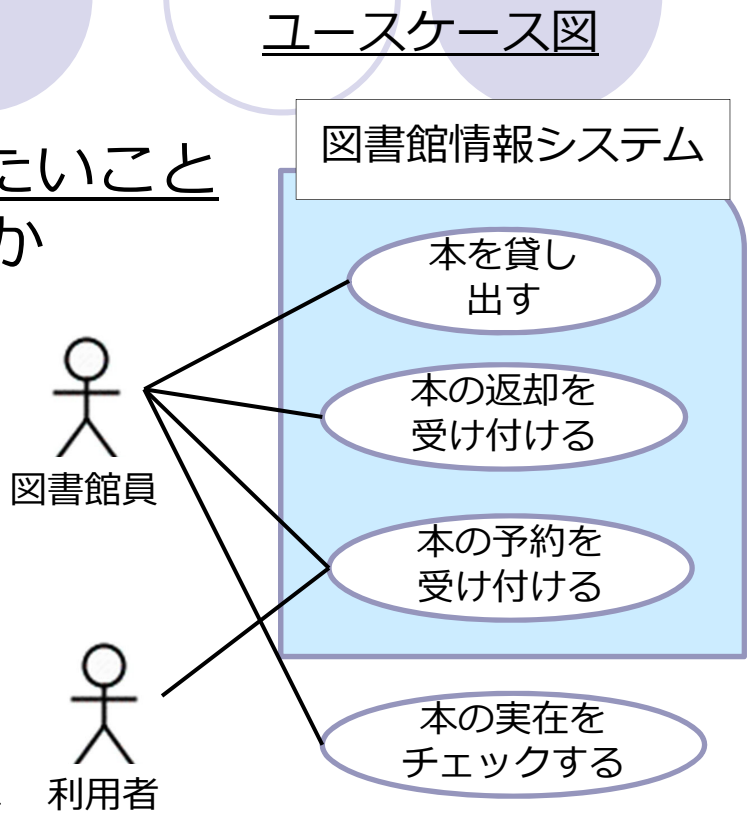


(a) 本の返却を受け付ける

(a) 名称：本を貸し出す
(b) アクター：図書館員
(c) 目的：誰にどの本をいつ貸したのか記録したい.
(d) 事前条件：記録なし
(e) 事後条件：記録あり
(f) 基本系列：
　(1) 図書館員がIDを与える
　(2) システムが貸出OKを表示する
　(3) システムが貸出を記録する
(g) 代替系列：貸出上限数を越える場合，貸し出しを拒否する.
(h) 具体的シナリオ例：．．．

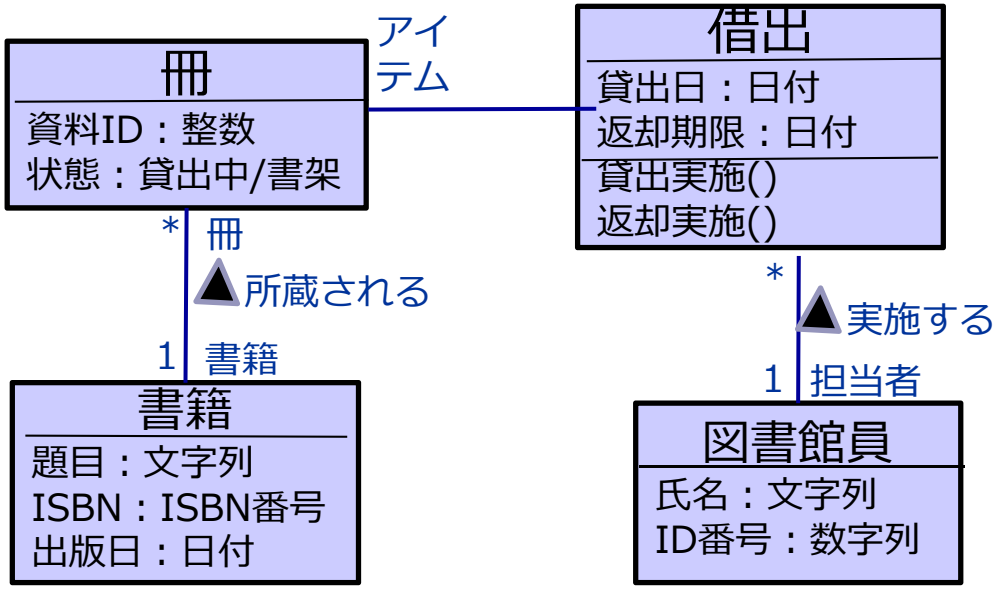
ユースケース図

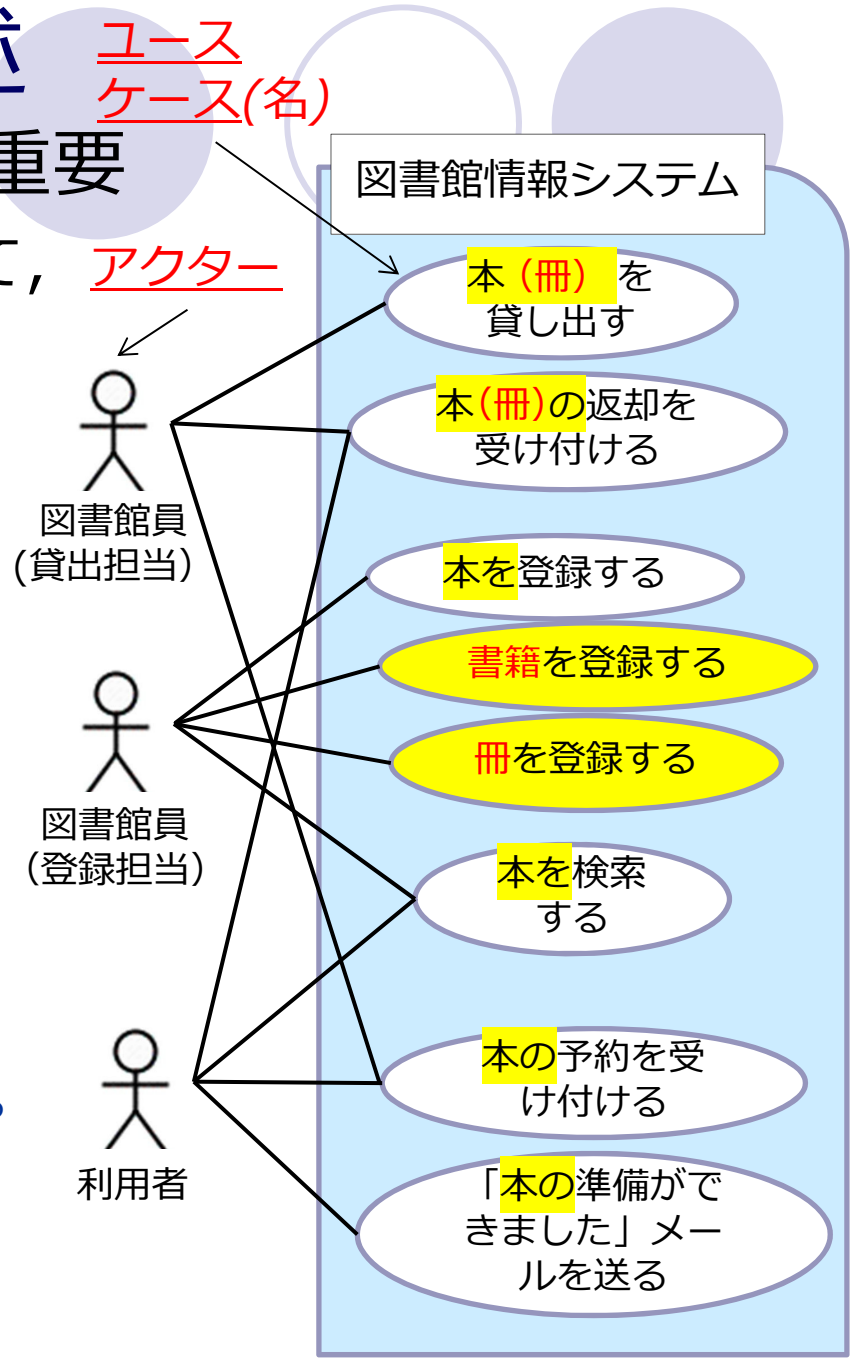
- ユースケース図が表すもの
  - アクターがシステムを用いてやりたいこと
    ≒ システムがその際になにをするか
    ≒ システムの「機能」
    - 「なにが」できるかを表す
    - 「どのように」ではない
  - 概念レベル
  - なには対象外かを明示化する
    ことにも使える
- 主な使い方：要求分析に用いる
  - 機能的な「要求仕様」 ＝システムが実現すべき機能
    - ユーザは要求をうまく表現できない
    - システム開発者と共同で明確にする
  - ユースケース図は明確にするための表現媒体



ユースケース図の記述

- 列挙し，明確化することが重要
  - クラス図に現れる概念を用いて，機能を明確にする
    - 例：貸し出しの実施者
    - 例：書籍と冊の区別
      - 貸出の対象は? 「冊」
      - 予約の対象は?
      - 検索時の情報と結果は？





ユースケースの内容の記述(1)

- 各ユースケースごとに内容を記述する
  - ※UMLでは規定されていない．簡略化したものを示す
- (a) ユースケース名（ユースケース図に現れるものと同じ）
  - そのユースケースでアクターがシステムを用いてしたいこと
  - 例：図書館員が本を貸し出す．利用者が書籍を予約する．
- (b) アクター名
  - 関わるユーザやシステム．例：図書館員（貸出担当）
  - 固有名は書かない．ユーザの「ロール名」を書く
    - システムの利用者（人間，組織，他システム）の役割
      - 同じ利用者でも，貸出手続き時は借出者，返却時は返却者
- (c) 「目的」
  - このユースケースでどのような変化を起こしたいか
    - 例：図書館員が（本の状況を管理するために）誰にどの本をいつ貸したのか記録したい．
  - (d), (e) の条件（状態）の変化で捉えると分かりやすい

ユースケースの内容の記述(2)

- (d) 事前条件
  - ユースケースの実行が行われる前の状態（事前状態）
  - 例：貸し出した事実が記録されていない
- (e) 事後条件
  - ユースケースの実行が行われた後の状態（事後状態）
  - 例：貸し出した事実(利用者，本(冊)，日付）が記録されている
  - (c)の目的は，一般に「事前条件を事後条件に変えること」
- (f) 「基本系列（シナリオ）」
  - システムとアクターの間の典型的な「対話」 （やりとり）の系列
  - 例
    - (1) 図書館員がユースケースを起動する
    - (2) システムが図書館員に利用者と本（冊）のID を要求する
    - (3) 図書館員が利用者IDと本（冊）のID を提示する
    - (4) システムは利用者が借出可能であることを確認する
    - (5) システムは貸出(利用者ID, 冊ID, 日付）を記録する

ユースケースの内容の記述(3)

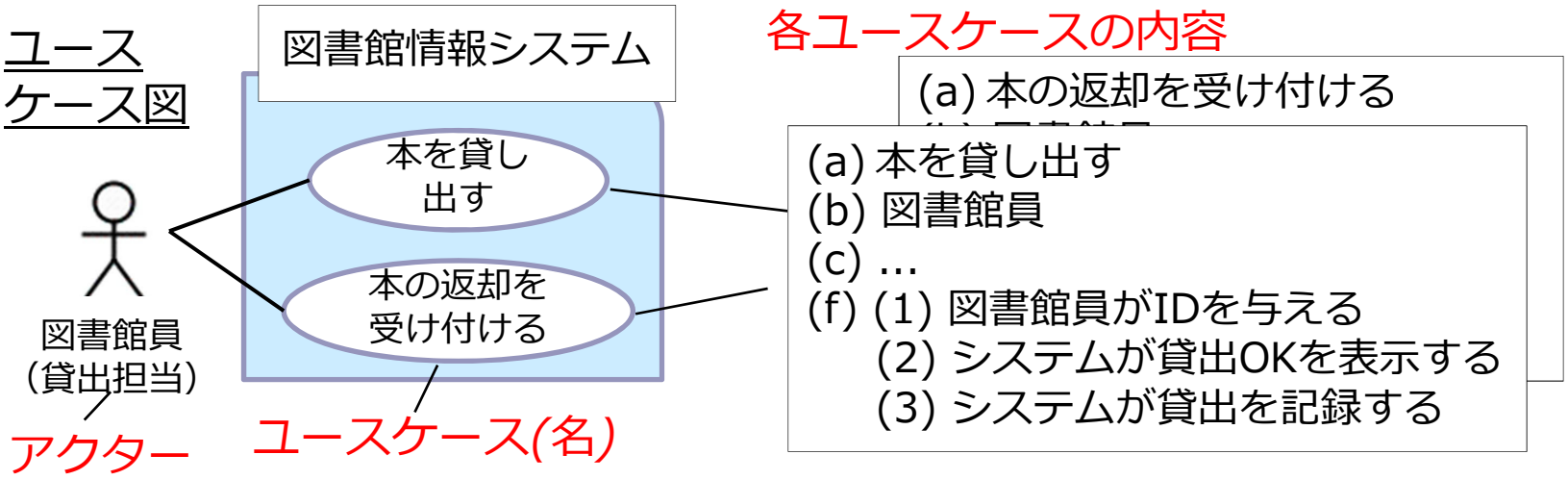
- (g) 代替系列
  - 例外的な（エラー時など）対話の系列
  - 例：基本系列の(4)で利用者が貸出上限数を越える場合，貸し出しを拒否する．(5)は行わない
- (h) 具体的シナリオ例
  - 具体的に，インスタンスが行う行為列を書いてみる．
  - エラーや例外的なケースも含むように書く
  - 例：
    - (ア) 來村が2019年4月24日に「鹿の王」を借りるために，貸出カウンターに来た．図書館員がシステムを起動する．システムが利用者IDと本IDを要求する．図書館員が來村のIDと，本のIDをシステムに読み込ませる．來村のIDは有効で，現在貸出数は0だったので，貸出処理は成功し，（UID0123, BID0789, 2019/4/24)という記録が残った．
    - (イ）.... 來村の現在貸出数が上限の10だったので，貸出は拒否された．
    - (ウ）.... 來村のIDは無効（期限切れ）だったので，拒否された．

ユースケースの内容の記述(4):注意点

- 書いてはいけないこと
  - システム内部の動作を書かない
    - あくまで外部的観点（≒ユーザ目線）からシステムを捉える
  - 「どうやって」目的／機能を達成するかを書かない.
  - ユーザインターフェイスを規定しない
    - 例：どうやって利用者IDを読み取るか
      - 磁気ストライプやバーコードなど.
- これらは「詳細設計時」に決めること
  - 概念設計段階では決めない
  - 大きくシステムが達成すべきことを決めることに集中する
- アクターやシステムの「対話」（やりとり）に注目する
  - どういう内容の情報（データ）がやりとりされるか
    - 例：利用者IDや本ID　（表現のされ方は問わない）
  - どういう「流れ」で対話が進むか
    - あまり細かい制御の流れは書かない．典型的なシナリオを中心に書く.

ユースケースのまとめ

- アクターがシステムを使用するシナリオ
  - アクター：システムの利用者のロール
  - ユースケース名：アクターの目的 ≒ システムの機能
    - ユースケース図で書かれる
  - ユースケースの内容：アクターとの「やりとり」を表す
    - システムがユーザによってどのように使われるか
- 目的：システムの外的な機能要求を明確にする

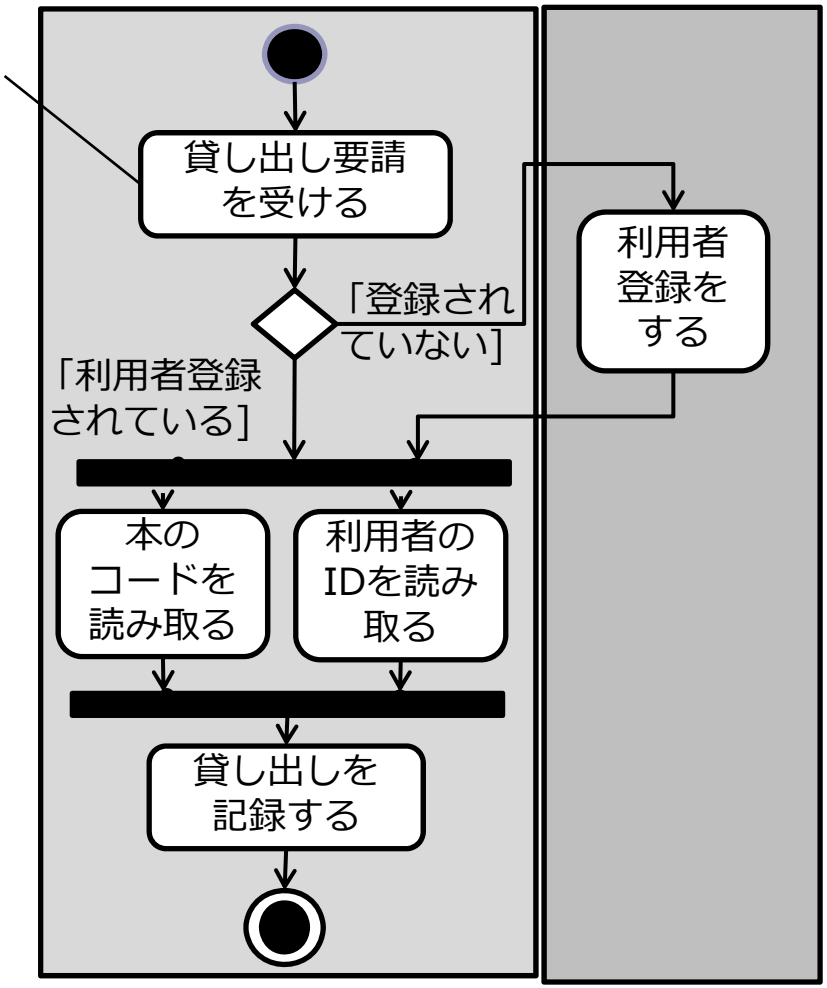


テーマと流れ（再掲）

- 外部的観点からの動的モデル
  - 外部的観点：システムを外部から捉える
  - 動的：時間を明示的に扱う
  - ユースケース図
    - 機能的要求，使用シナリオ
- 内部的観点からの動的モデル
  - アクティビティ図
    - 内部的観点から「処理（オブジェクトの動作）の流れ」を表す
    - 外部的観点からの「ワークフロー」の表現にも使われる
  - 次 シーケンス図
    - アクターやオブジェクト間の「対話」を表す
    - メッセージパッシング

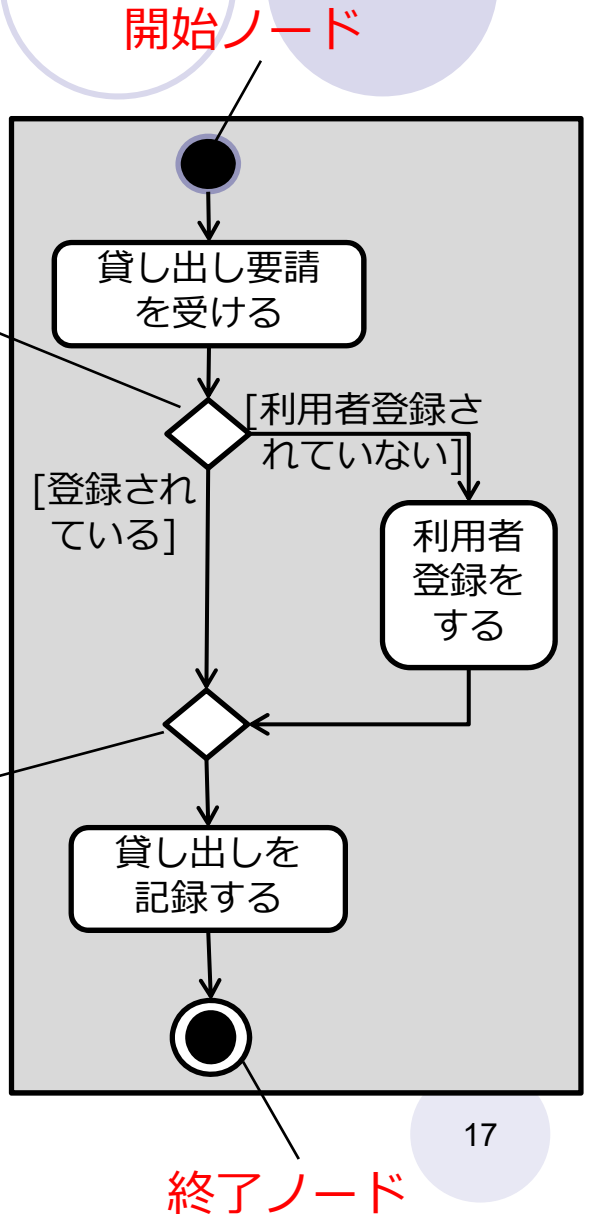
アクティビティ図

- 「アクション」
  - オブジェクトのひとつの活動／処理／動作
  - 振る舞いと呼ばれる
  - 角丸長方形で示す
- 「アクティビティ」図
  - アクションの間の時間的流れ（制御の流れ／プロセス）を表す
  - 基本：ひとつのアクションの動作が完了したあと，次のアクションの動作が開始される
  - 順序を実線の矢印で示す
    - 「制御フロー」と呼ばれる
  - アクションの「系列」（「アクティビティ」と呼ぶ）を表す
  - 処理の分岐や合流，並列なども表現できる.



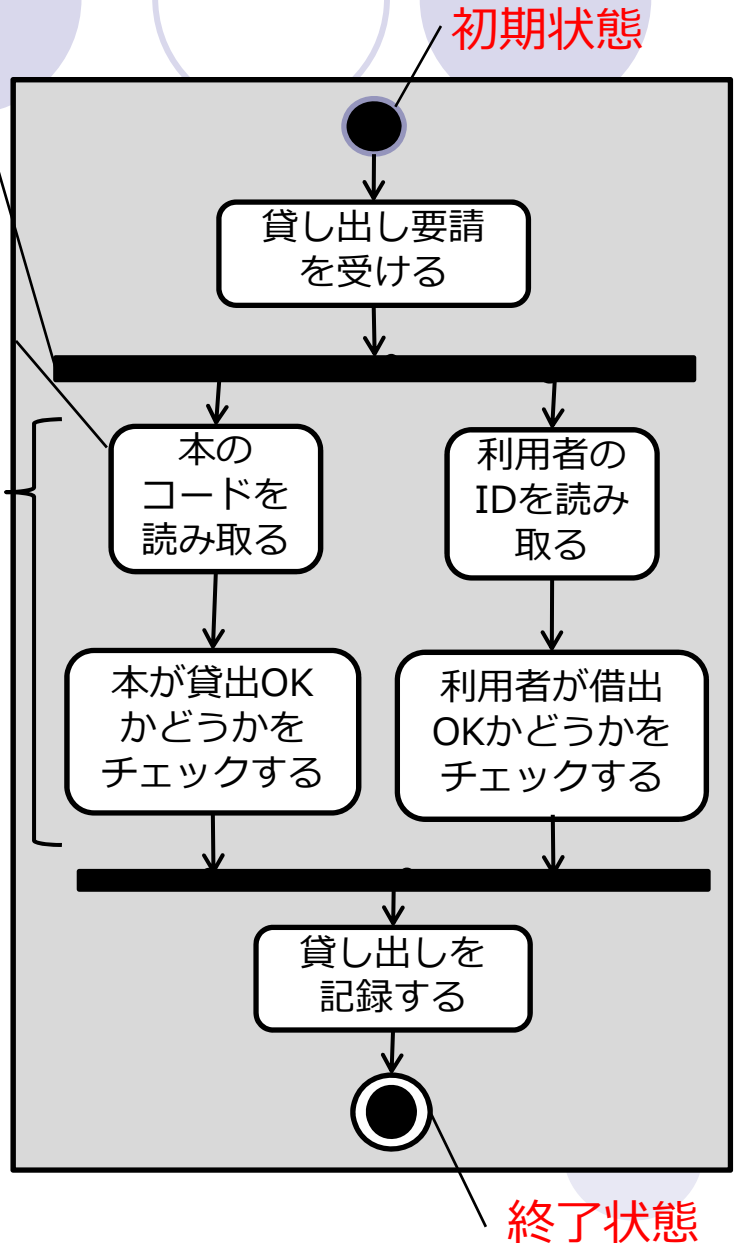
アクティビティ図の要素(1)

- 開始ノード／終了ノード
  - 開始ノードから終了ノードへ処理が流れる
- 「ディシジョン」ノード
  - 状態によって処理の流れが分岐する
  - 「菱形」で分岐点を表す
  - １つの入力フローと，複数の出力フロー
  - 出力フローには条件(ガード)を書く
    - [ ] 内に論理式を書く
    - ガード条件は排他的OR (XOR)
    - [else]は，他の全ての条件が偽であるときに真となる，ことを表す．
  - 実行条件付きのアクション系列（アクティビティ）を表す
- 「マージ」ノード
  - 判断（分岐）によって生じた条件つき振る舞いの終了を表す
  - 菱形．複数の入力フローと１つの出力フロー



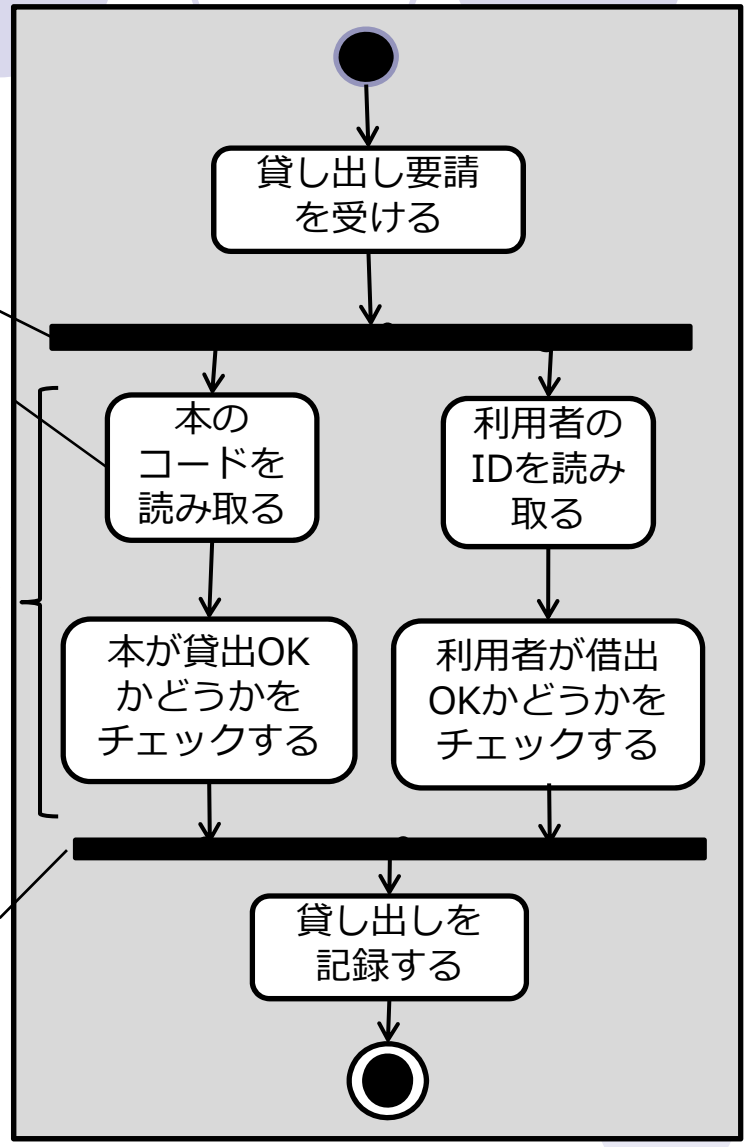
アクティビティ図の要素(2)

- 並列動作の表現
- 「フォーク」ノード
  - 以下のアクティビティの実行が「並列的」でもよいことを表す
  - 太い棒線で表す
    - １つの入力フローと複数の出力並行フローを持つ
  - 次の太い棒線（ジョイン）までの複数のアクション系列（アクティビティ）を表す
  - 各系列内のアクションの順序のみを規定する．系列の間の実行順序は規定されない（どちらが先でもよい）
  - 守るべき本質的な順序だけを規定する．(規定されていない順序は実装時または実行時に決めてよい)



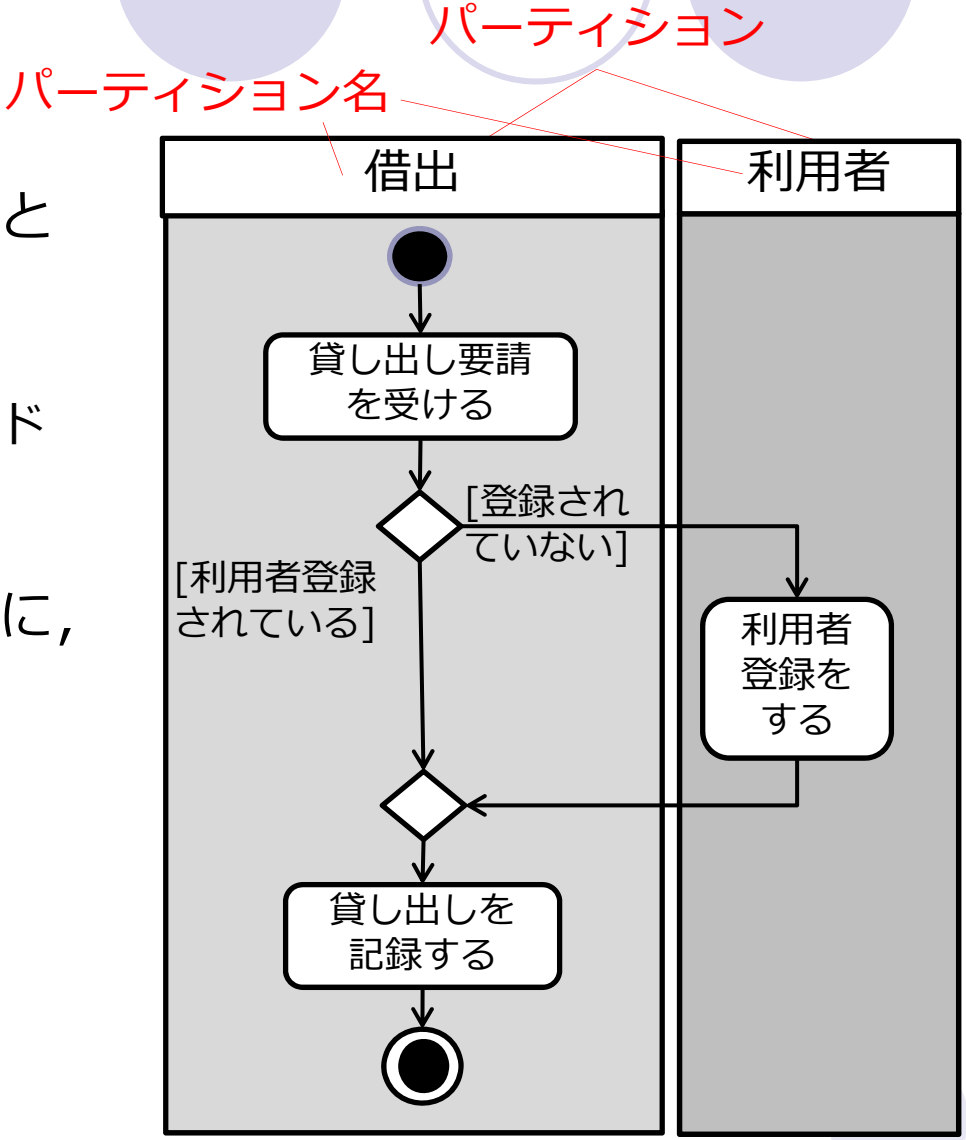
アクティビティ図の要素(3)

- 「ジョイン」ノード
  - 太い棒線で表す
    - 複数の入力フローと
      １つの出力フローを持つ
  - 複数のアクション系列
    （アクティビティ）の
    流れをまとめる
  - 全てのアクション系列の
    実行の終了を待つ.
  - 全てが終了したときに,
    出力側フローが実行される.
  - 「実行の同期化」と呼ばれる.
- 並列アルゴリズムの実装
  - 複数スレッドの開始を
    フォークで表す
  - 実行の同期をジョインで表す



アクティビティ図の要素(4)

- 「パーティション」
  - アクティビティの実行主体となるアクター，または，オブジェクトを表す
    - 例：「利用者登録」メソッドは利用者オブジェクトが処理する．
  - 複数の実行主体がある場合に，区別するために用いる．

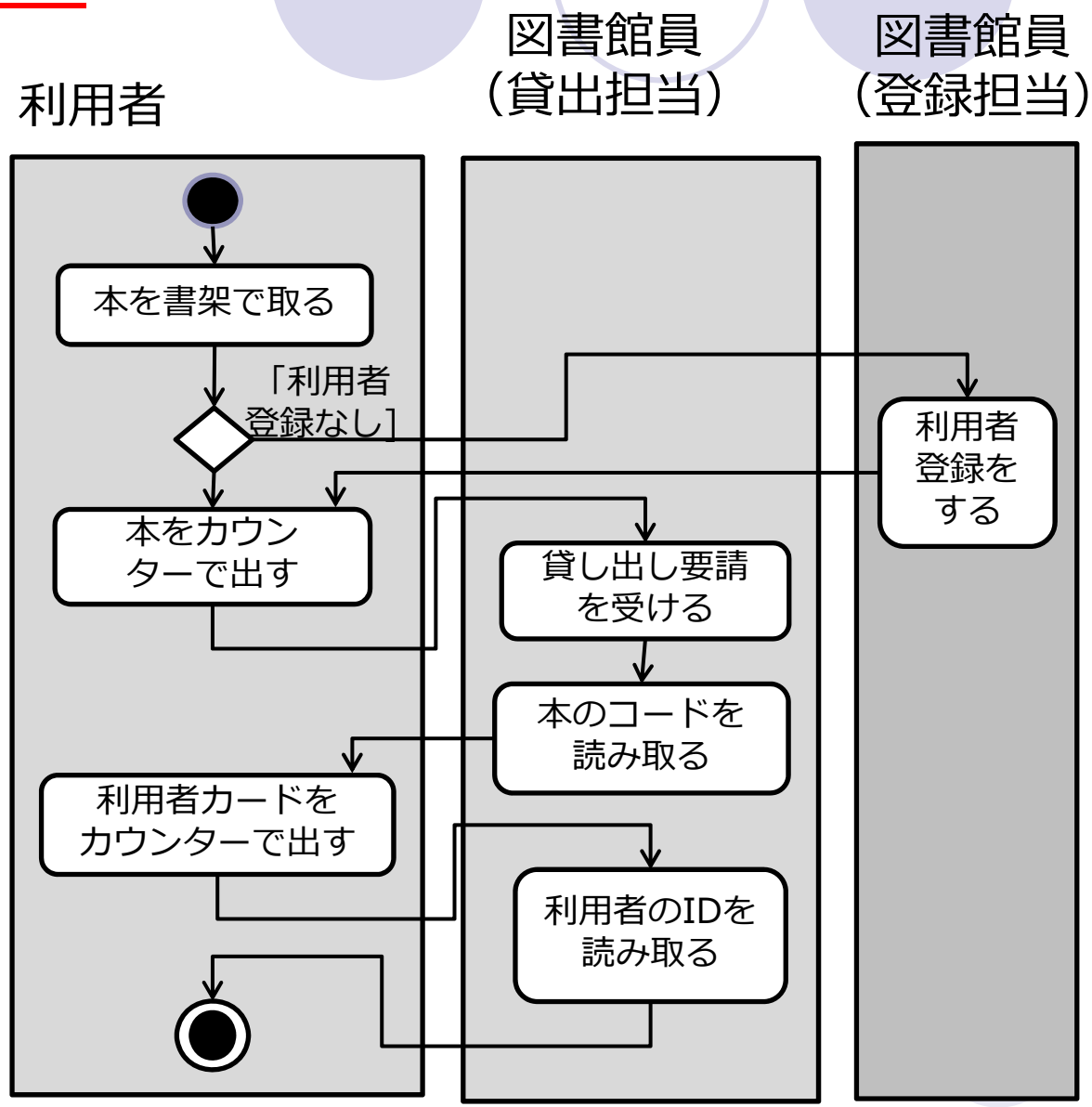


アクティビティ図の用途

- (1) システムの内部的な処理の流れを表す
  - 内部的観点
  - 情報システムが内部的にどう動くかを表す
  - オブジェクト間の処理の流れの時間的タイミングを表す
  - フローチャートに似ている
  - 設計フェイズの後半の詳細設計で用いられる
- (2) 対象世界のワークフローを表す
  - 外部的観点
  - 概念設計で用いる
  - 各アクションは対象世界で人や組織が行っている仕事や処理を表す
  - 仕事の流れ（ワークフロー）

「ワークフロー」の表現

- 処理の流れと分担を表す
  - 区画はアクター（人）を表す
  - 組織の場合もある
  - アクションは人の行為を表す



まとめ

- 外部的観点からの動的モデル
  - 外部的観点：システムを外部から捉える
  - 動的：時間を明示的に扱う
  - ユースケース図
    - アクターがシステムを使用するシナリオ
    - システムの機能，アクターとのやりとり
- 内部的観点からの動的モデル
  - アクティビティ図
    - 内部的観点から「処理（オブジェクトの動作）の流れ」を表す
    - 外部的観点からの「ワークフロー」の表現にも使われる
  - 次 ⇨ シーケンス図
    - アクターやオブジェクト間の「対話」を表す
    - メッセージパッシング

参考文献

- [1] 児玉公信（著），UMLモデリングの本質，日経BP社，2004
- [2] マーティン・ファウラー（著），羽生田 栄一 (翻訳)，UMLモデリングのエッセンス（第３版），翔泳社，2005
- [3] かんたんUML入門 改訂2版，技術評論社，2017

- ※特に図書館情報システムのUML図は [1] を参考にして，改変した．ユースケースの内容の記述は，UMLで規定されていないため，[1]をベースにしつつ，[2]の知見と[3]の最新の規定を取り入れて，大幅に改変した．