ソフトウェア設計法及び演習
ソフトウェア工学概論及び演習

関澤 俊弦
日本大学 工学部

連絡

- 設計演習2（来週の7月4日）
  - 出題範囲
    - オブジェクト指向開発
      （主にLesson06~Lesson10）
  - 設問形式
    - WordとAstah*を用いたレポート形式
  - 提出期限
    - 設計演習2の講義日に案内
      （組により提出期限が異なる可能性あり）

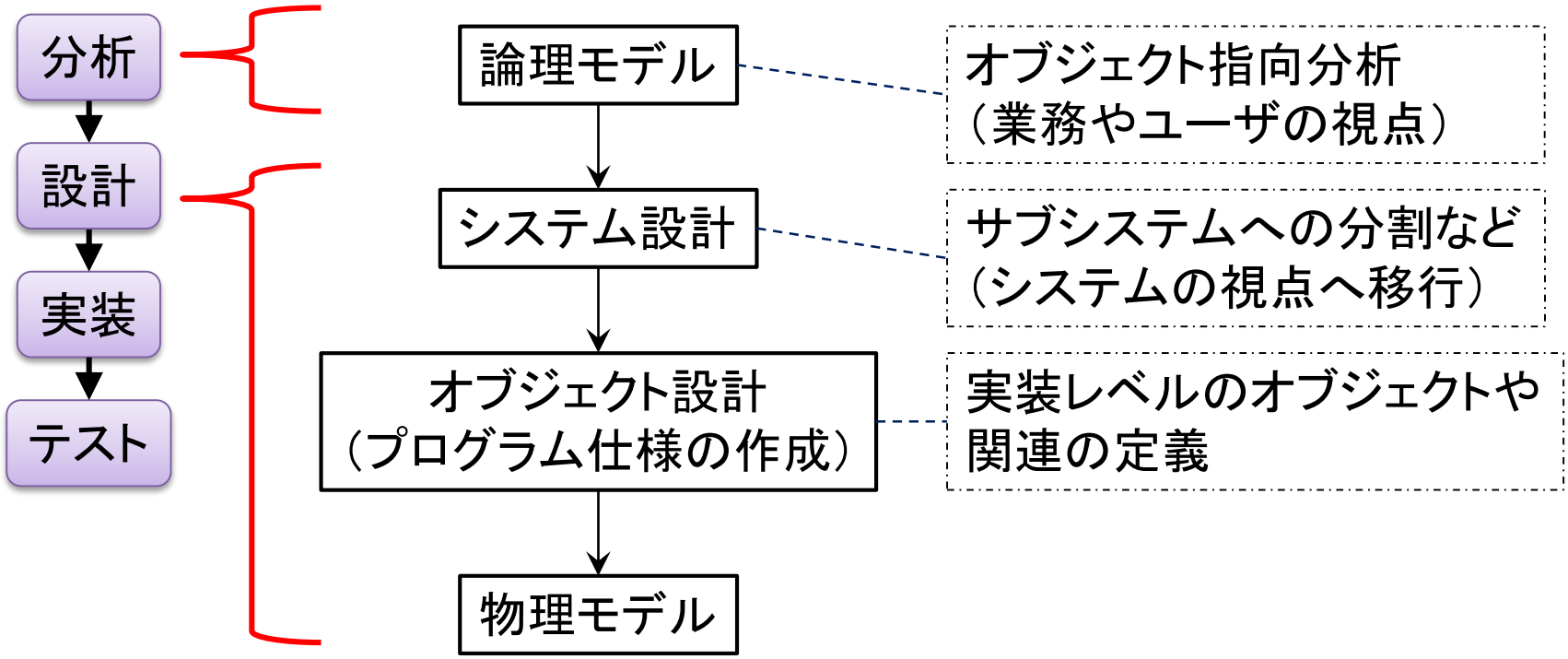
復習

- オブジェクト指向
  - データ属性，メソッド
  - クラス，インスタンス
- オブジェクト指向によるシステム分析
- UML
  - ユースケース，クラス図，シーケンス図，ステートチャート

- オブジェクト指向によるシステム設計
- 構造化設計
- 演習

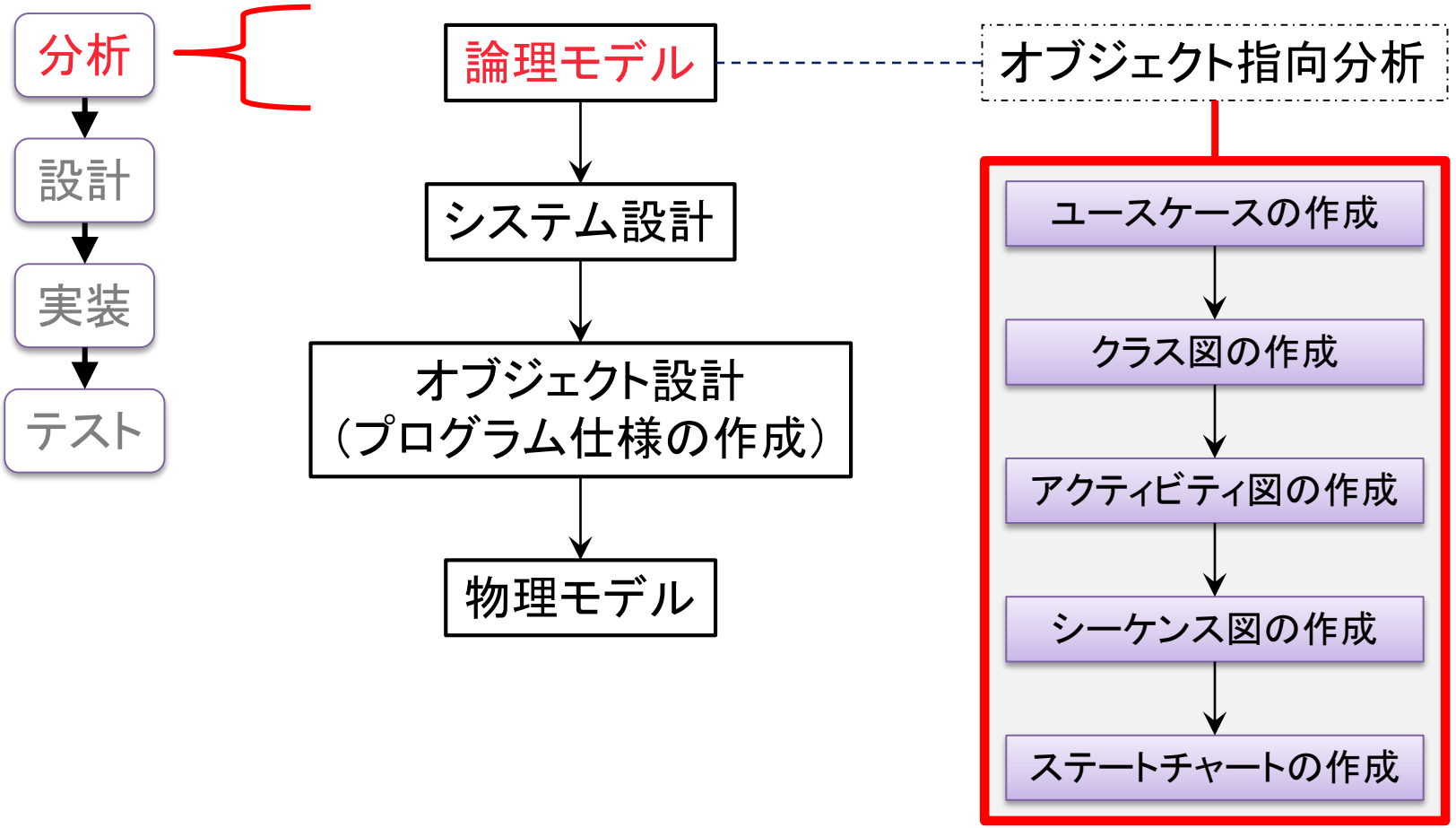
オブジェクト指向によるシステム設計

- オブジェクト指向モデリング
  - 分析段階で**業務やユーザの視点**のモデル
    → 設計段階で**システムの視点**のモデル



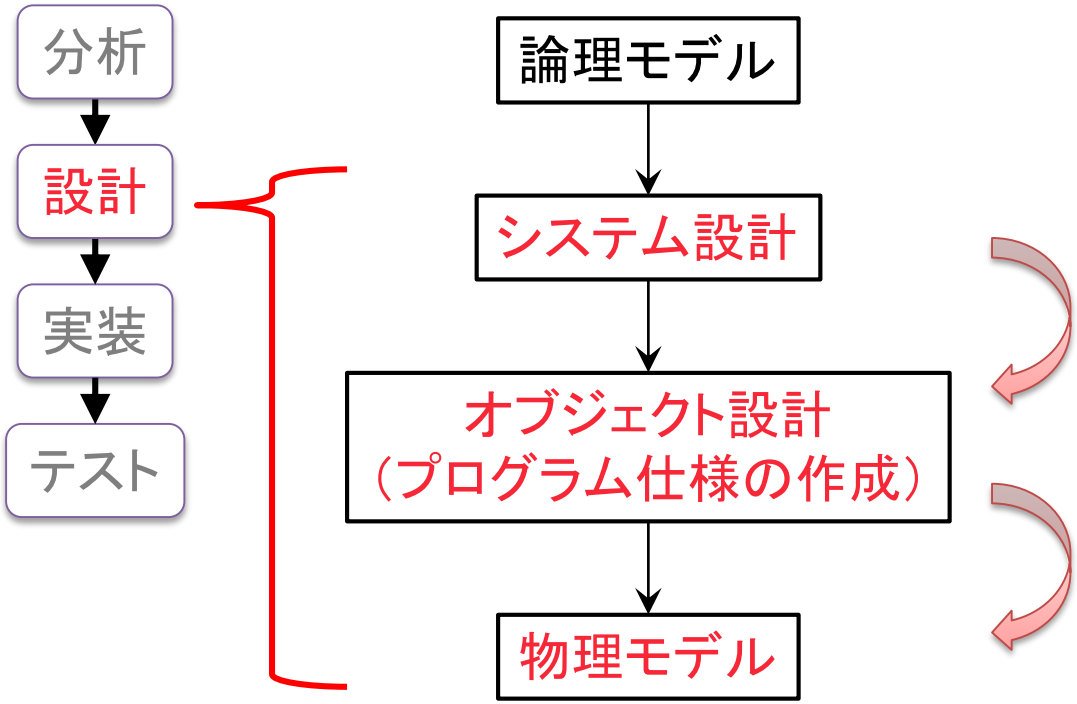
オブジェクト指向によるシステム設計

- 分析段階 (復習：Lesson07，Lesson08)



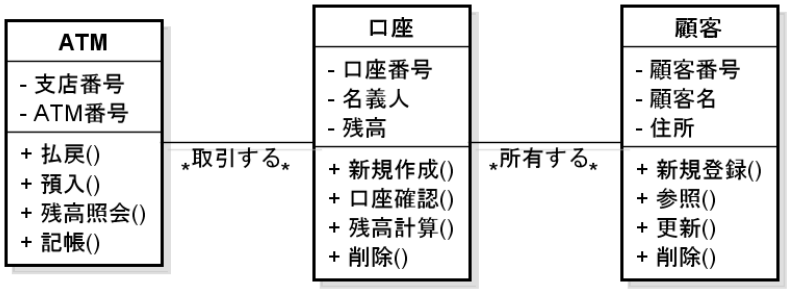
論理モデルの展開

- 設計段階
  - 物理モデルへ展開
    - 実装言語やデータベースに合わせた実装用のクラスの表現



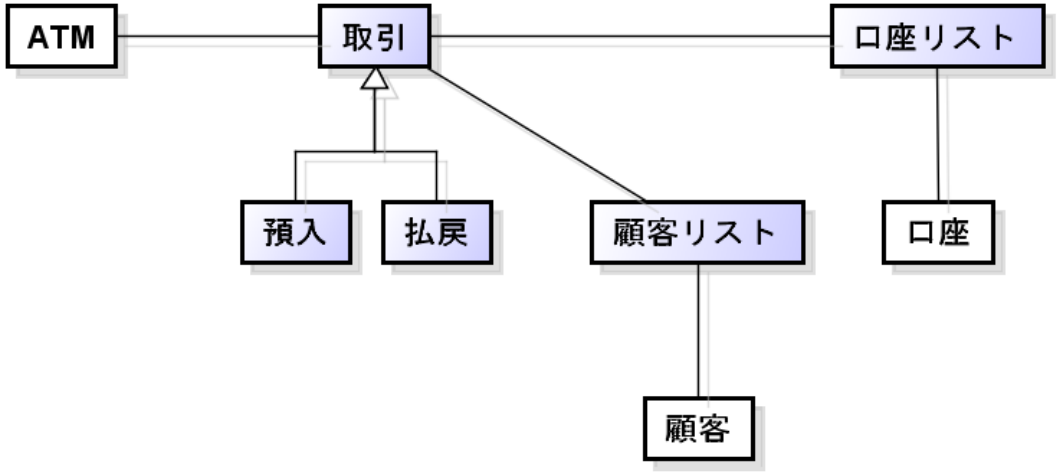
例: ATMの論理モデルの展開

- 論理モデル



p.96

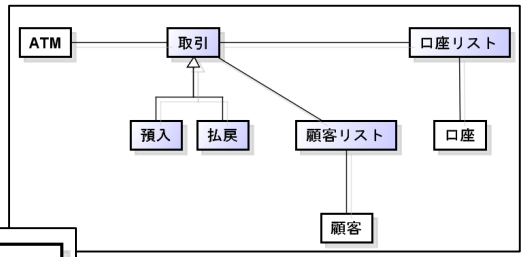
- 物理モデル
  - 実装用のクラスの追加

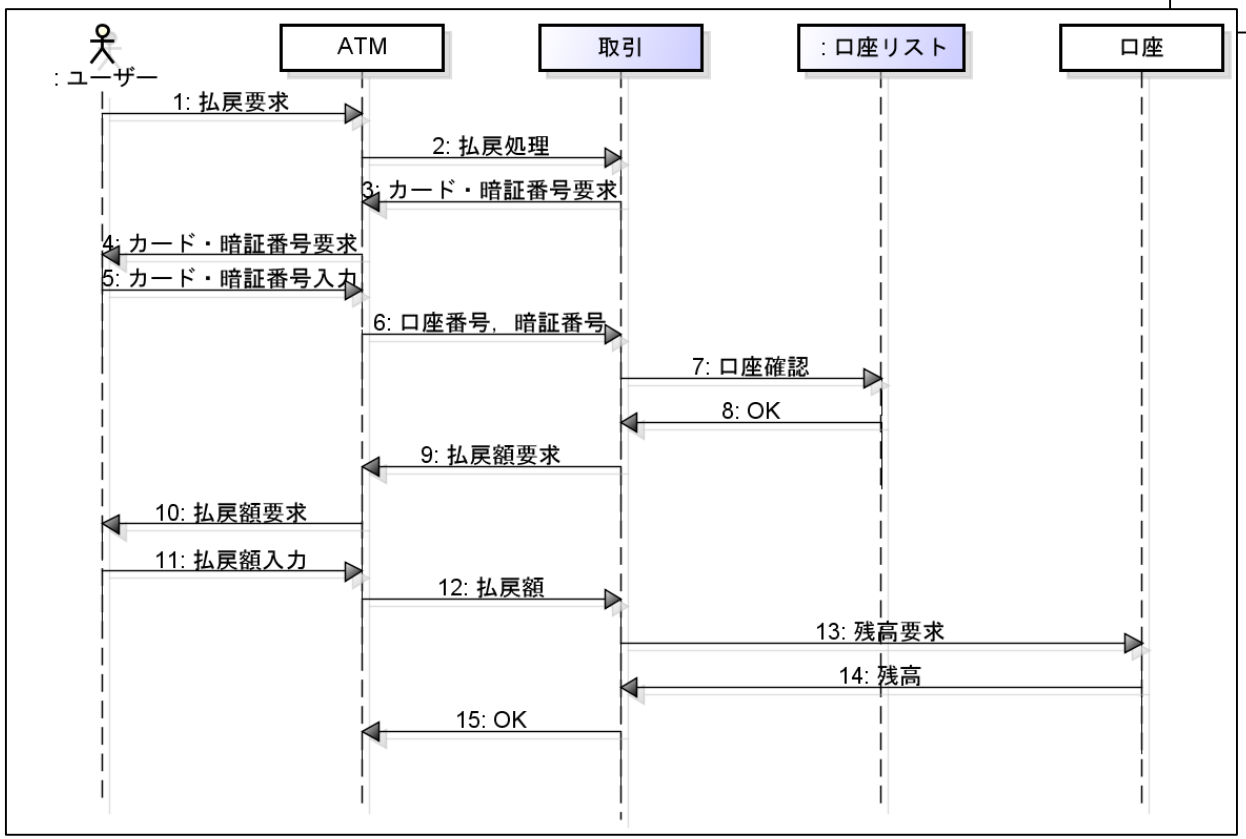


p.105

例: ATMの論理モデルの展開

- 物理モデルのシーケンス図
  - 実装用のクラスの追加

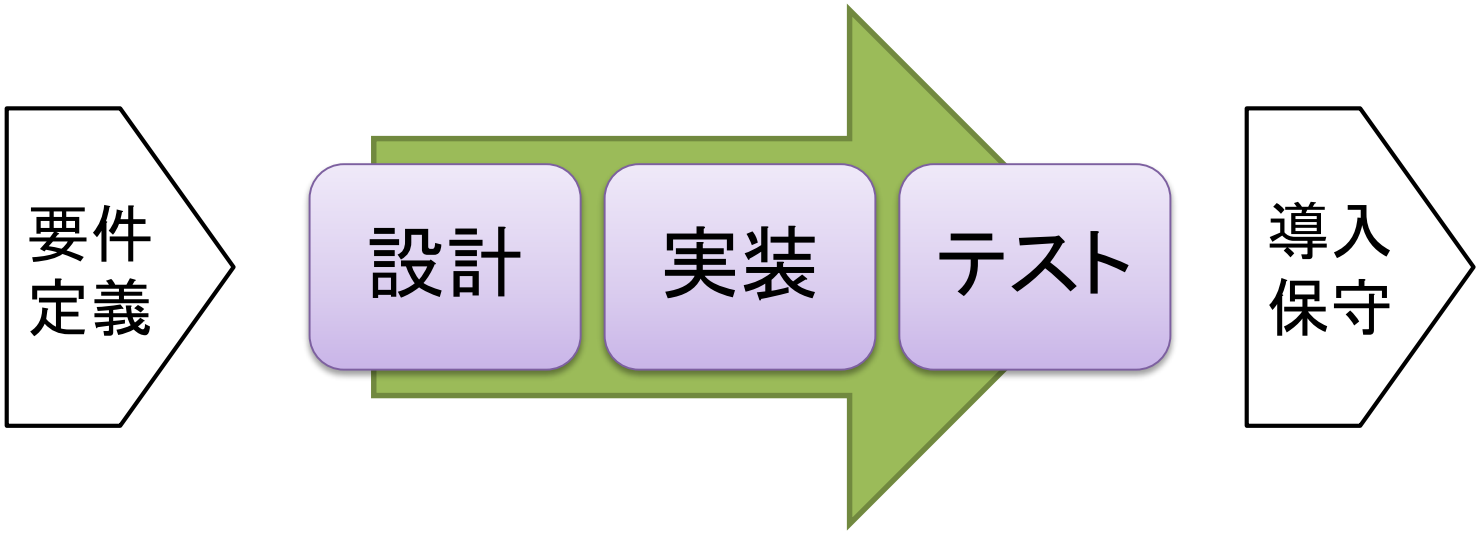




- オブジェクト指向によるシステム設計
- 構造化設計
- 演習

設計フェーズ

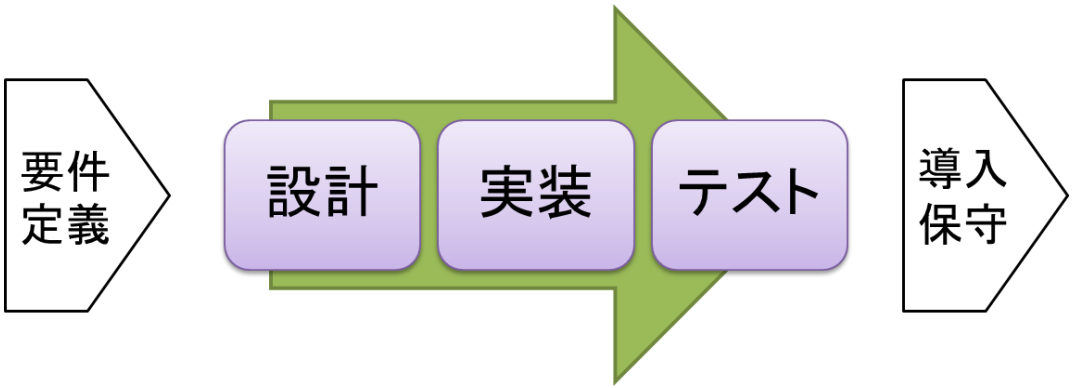
- システムの設計フェーズ
  - 一般に，外部設計と内部設計から成る
    - 外部設計: 業務やユーザの視点で捉えた設計
    - 内部設計: 計算機の視点で捉えた設計



構造化設計

- 構造化設計とは
  - 設計段階で効果的な手法

- 「よく」管理されたプロジェクト
  - 設計の各過程が明確
  - 方法論・品質測定基準などが明確



システム設計における問題点

- 方法論の欠如
  - 設計に関する標準的な方法論がない(?!)
- プログラム至上主義
  - システム要件や仕様に不明確を残した状態で，コーディングに入ってしまう
    - 「プログラム」のアウトプットがある安心感
- 修正作業の増加
  - 設計不足による手戻りの増加


設計の「複雑さ」がおおきな原因の1つ


p.116

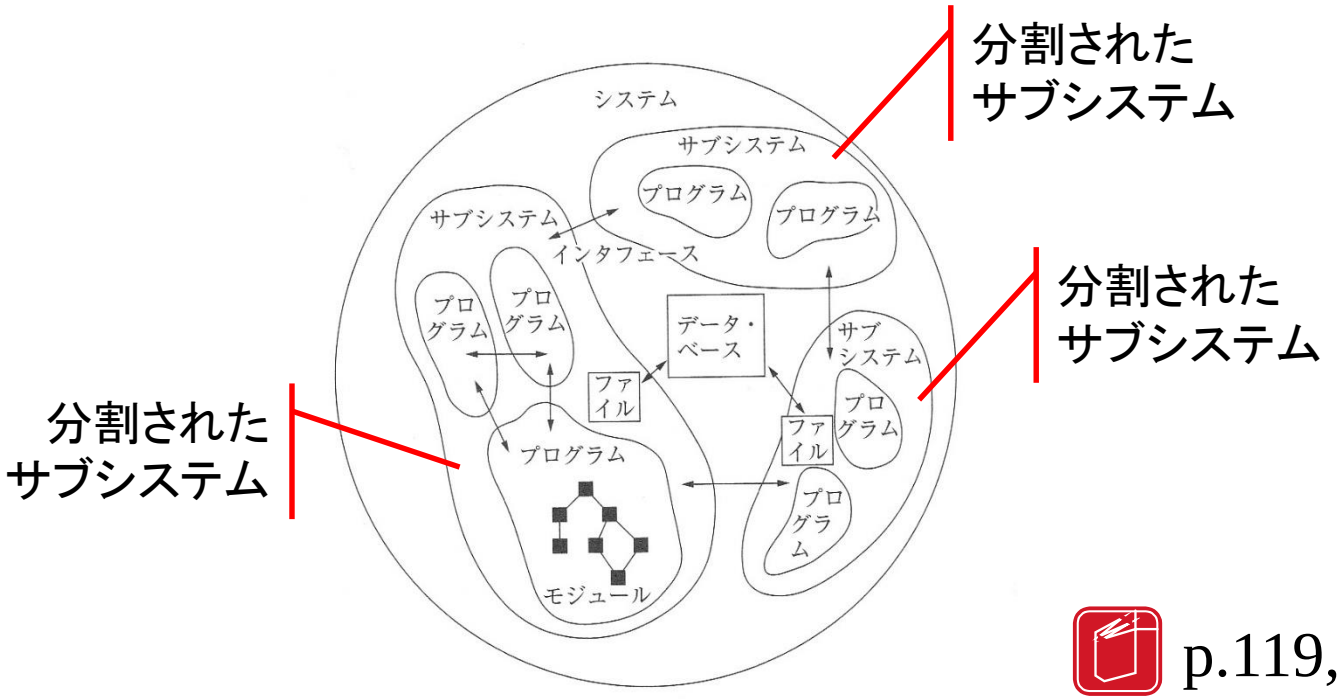
「良い設計」の概念

- 複雑さの最小化
  - 分割
  - 独立性
  - 強度
  - 結合度
  - 階層化
- 用語: モジュール
  - システムの構成要素となるもの
  - 通常，いくつかの機能を集め，まとめる


p.118~

複雑さの最小化: 分割

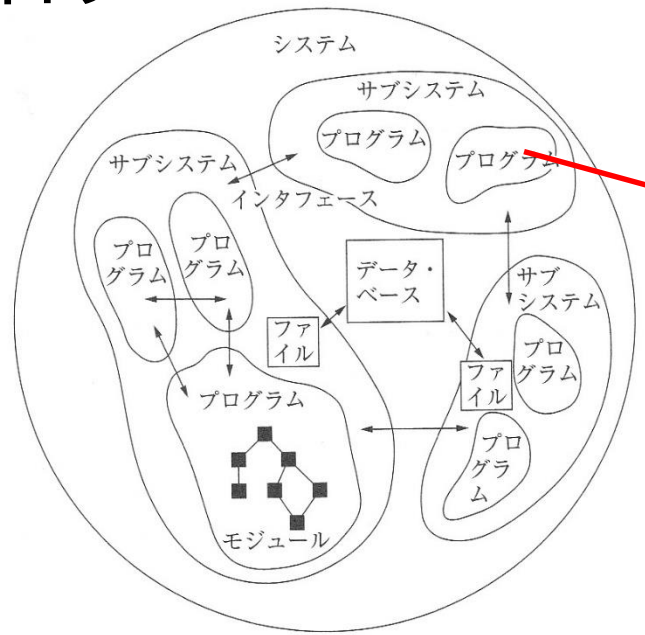
- システムを構成要素に分割すること
  - 問題を局所化し，複雑さを低減
  - 抽象化と階層化が必要



p.119, 図7.1

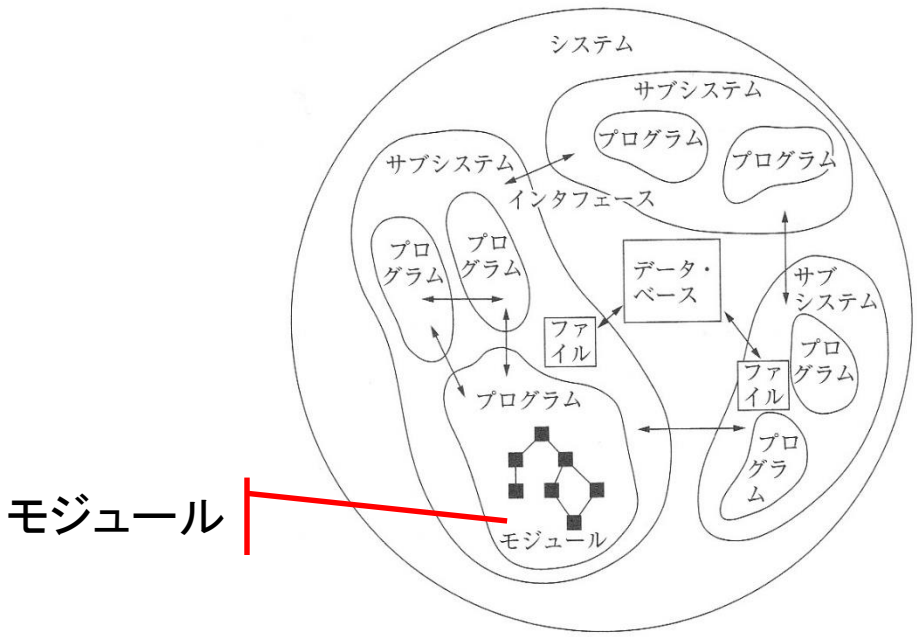
複雑さの最小化: 独立性

- システム/プログラムの構成要素の独立性のこと
  - 他の構成要素との関連を単純化するほど，独立性が高いと言う



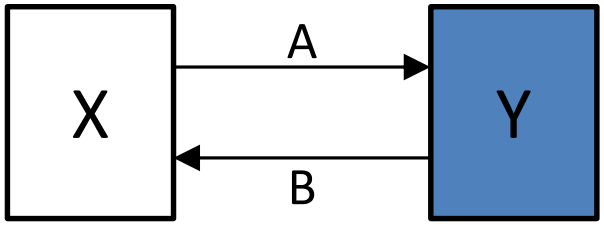
複雑さの最小化: 強度

- モジュールの強度とは
  - モジュールを構成する要素間の関連性の強さ
    - 個々の命令の必然性が高い程，強度は強いと言う



複雑さの最小化: 結合度

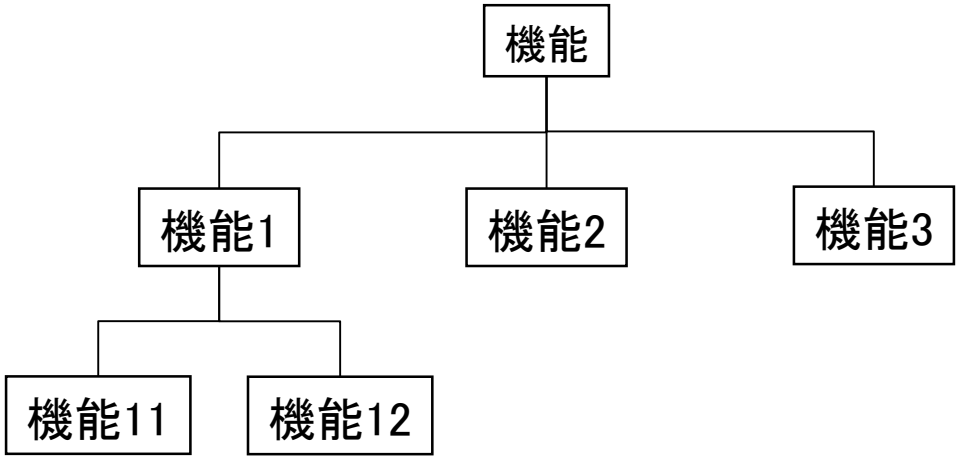
- モジュール間の関連性の尺度
  - 他のモジュールへの影響が少ない程，結合度は弱いと言う
    - 例: ブラックボックスは結合度が弱い
      ∵入出力のみが分かっていれば扱える



Xから見てYはブラックボックス
（Yの内部処理を知る必要がない）
（入力Aに対して出力Bが分かればよい)

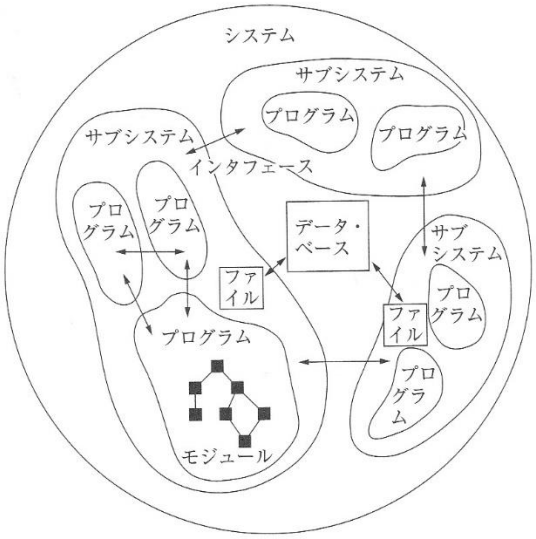
複雑さの最小化: 階層化

- システム/プログラムをいくつかのレベルに階層的に分割すること
  - 個々のレベルの理解を容易にする
  - 各レベルは下位のレベルの要素間の関連性を集約する



構造化設計: 要約

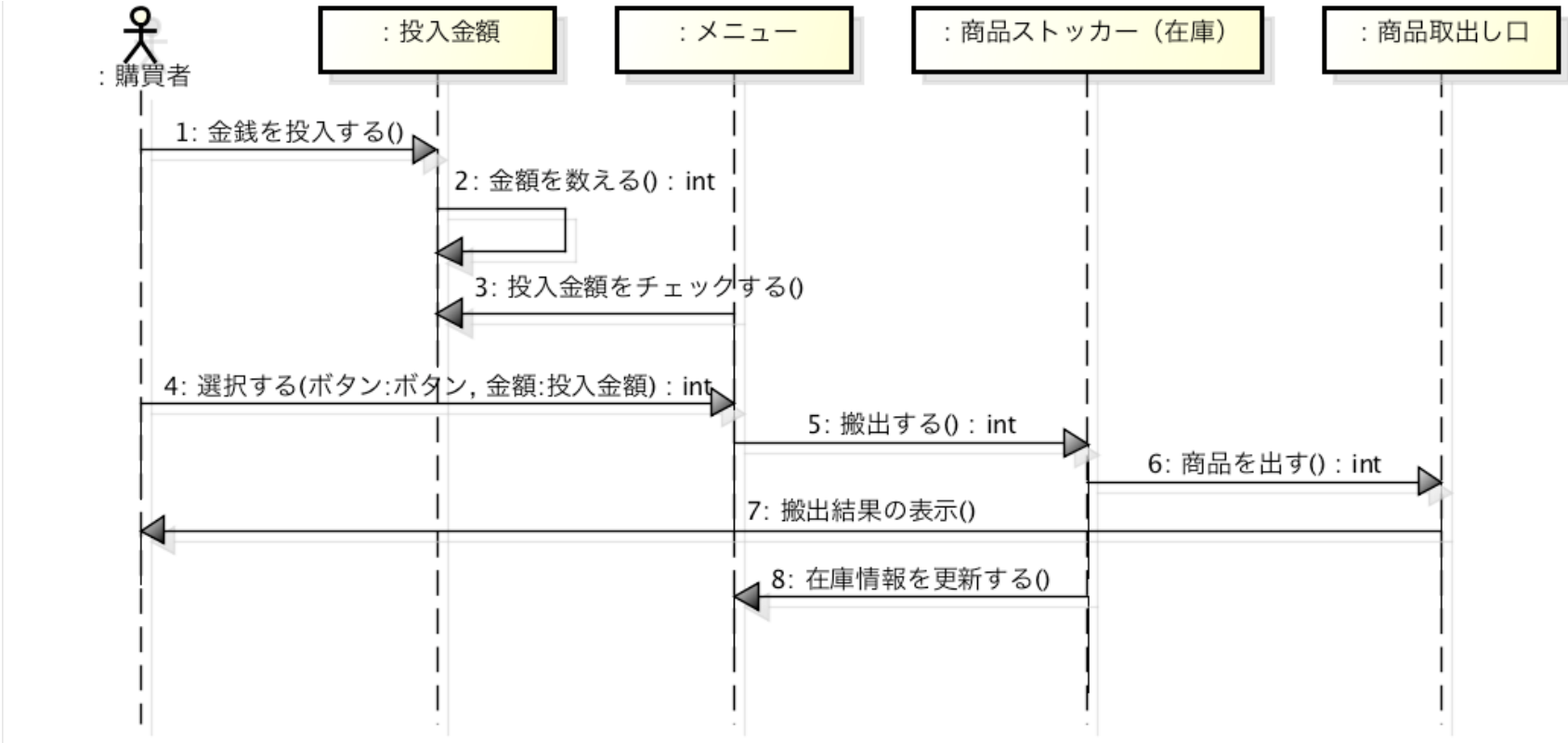
- 構造化設計では，システム/プログラムを;
  - 機能的に捉えて分割し（強度を強くし），
  - 機能の階層化を図り，
  - 個々の機能間の結合度を弱くする.
    （機能間の入出力を定義することにより）



- オブジェクト指向によるシステム設計
- 構造化設計
- 演習

演習9(前回): シーケンス図 - 解答例

- 自動販売機のシーケンス図をastah*で描け
  - イベントや操作呼び出しのシーケンスはシナリオと合っていること
  - クラス名はクラス図と一致していること



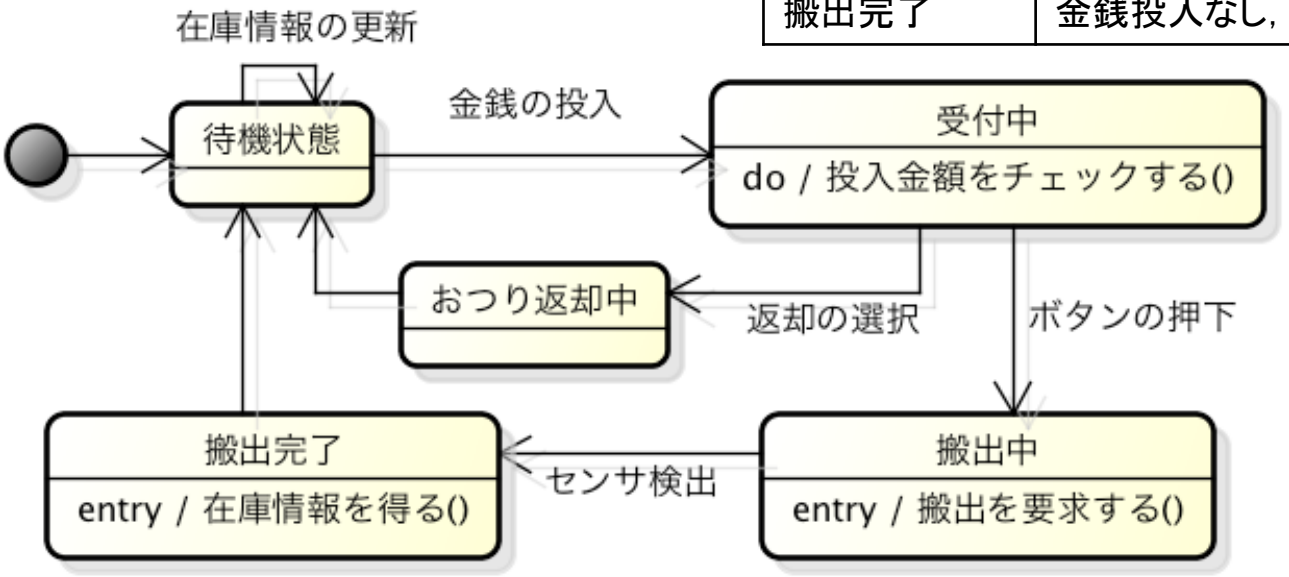
演習9(前回): ステートチャート - 解答例

- ステートチャートを描け

  メニュークラスの例:

  - オブジェクトの振舞い(イベントに応じた状態遷移)が正しく説明できること

状態名	内容
待機状態	初期状態，金銭投入なし
おつり返却中	金銭投入あり，おつりの返却
受付中	金銭投入あり， 選択可能なボタンの点灯
搬出中	金銭投入あり，搬出待ち
搬出完了	金銭投入なし，搬出済み



演習10: 自動販売機の分析

- 前回までに作成した自動販売機のクラス図やシーケンス図に合わせて，以下のモデルを修正し，完成させよ
  - ユースケース図
  - クラス図
  - シーケンス図
  - ステートチャート（1つ以上のクラス）

提出

- 提出物
  - 次のファイルを提出（詳細は後述）
    1. Astah*のファイル 1つ
- 提出先
  - ポータルサイトの「課題管理」
    - 提出期限までは再提出可能
- 提出期限
  - 次回講義の前日 23:59

提出物の詳細: Astah*のファイル

- ファイルフォーマット
  - Astah*のプロジェクトファイル
    - 1つのプロジェクトファイルにまとめること
      （ポータルは1ファイルのみ提出可能）
- ファイル名
  - "学籍番号_10.asta"
    - 拡張子は .asta とすること

まとめ

- オブジェクト指向
  - データ属性，メソッド
  - クラス，インスタンス
- オブジェクト指向によるシステム分析
- UML
  - ユースケース，クラス図，シーケンス図，ステートチャート
- オブジェクト指向によるシステム設計
- 構造化設計