

オブジェクト指向開発論

2020年5月21日

海谷 治彦

目次

- モデリングツールのインストール確認
- モデリングとは？
- モデリングの重要性
- 要求と仕様の違い
- オブジェクト指向 VS 従来型 (手続き型)
- 実例

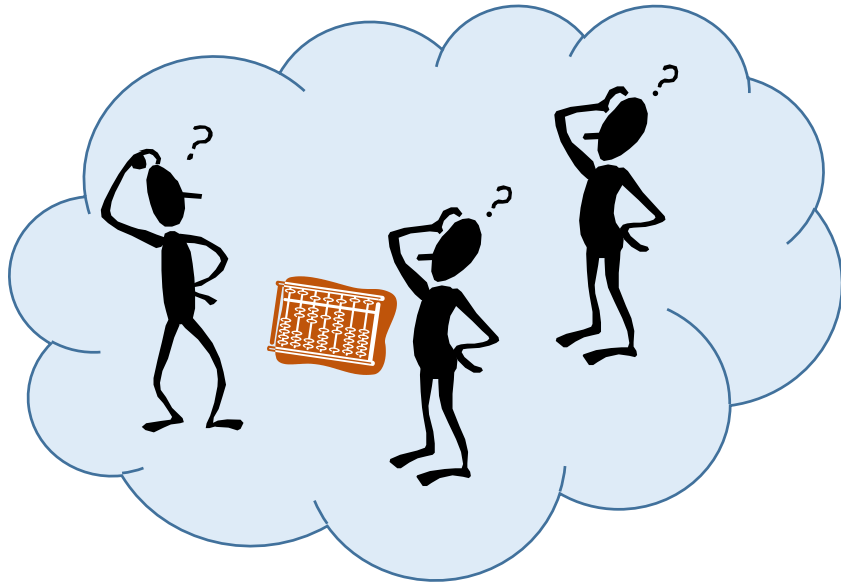
ツールのインストール確認

- 各自確認してください.
- 管理者権限なしでのインストール方法も以下で述べてます.
 - Teamsにもzipをおいてあります.

<https://www.sci.kanagawa-u.ac.jp/info/kaiya/astah/>

- 一応, メニュー等の英語/日本語切り替えも上記で述べてます.

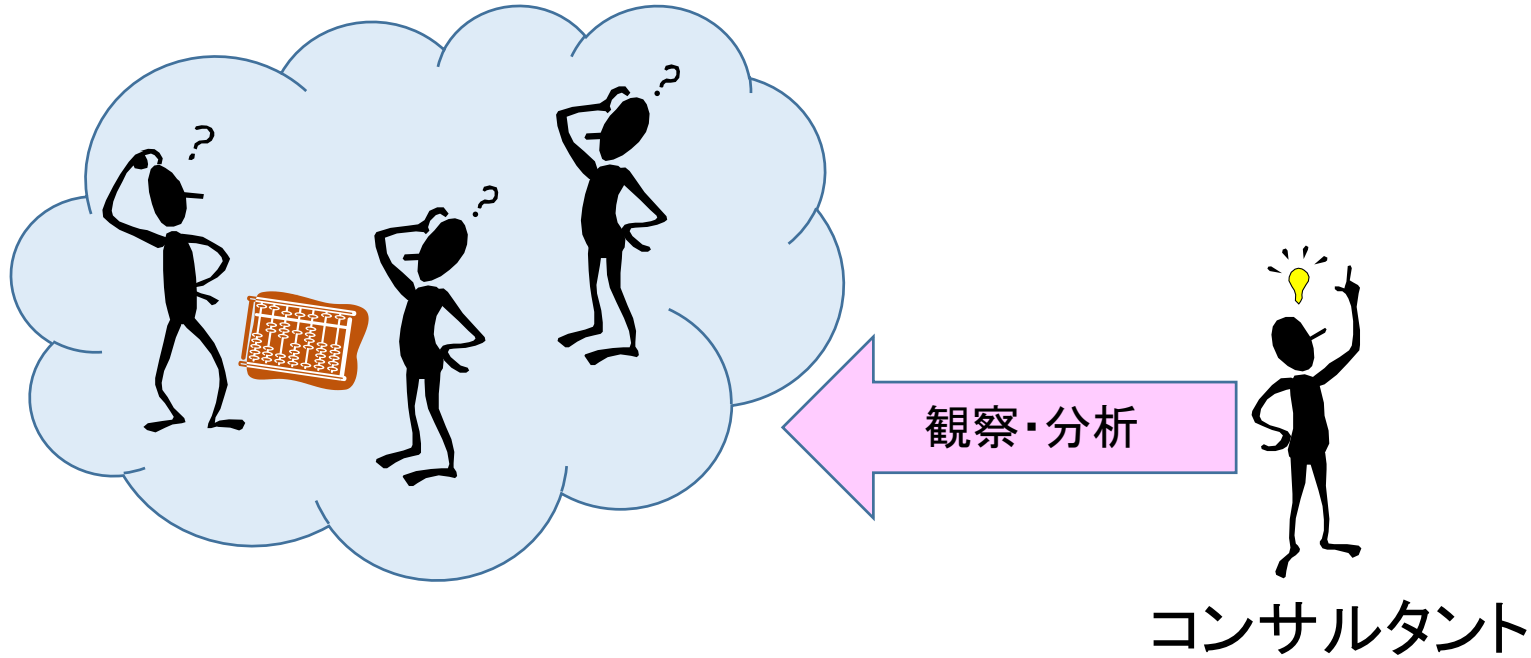
モデリングとは 1/4



仕事や生活と、
それにかかわる人々や道具

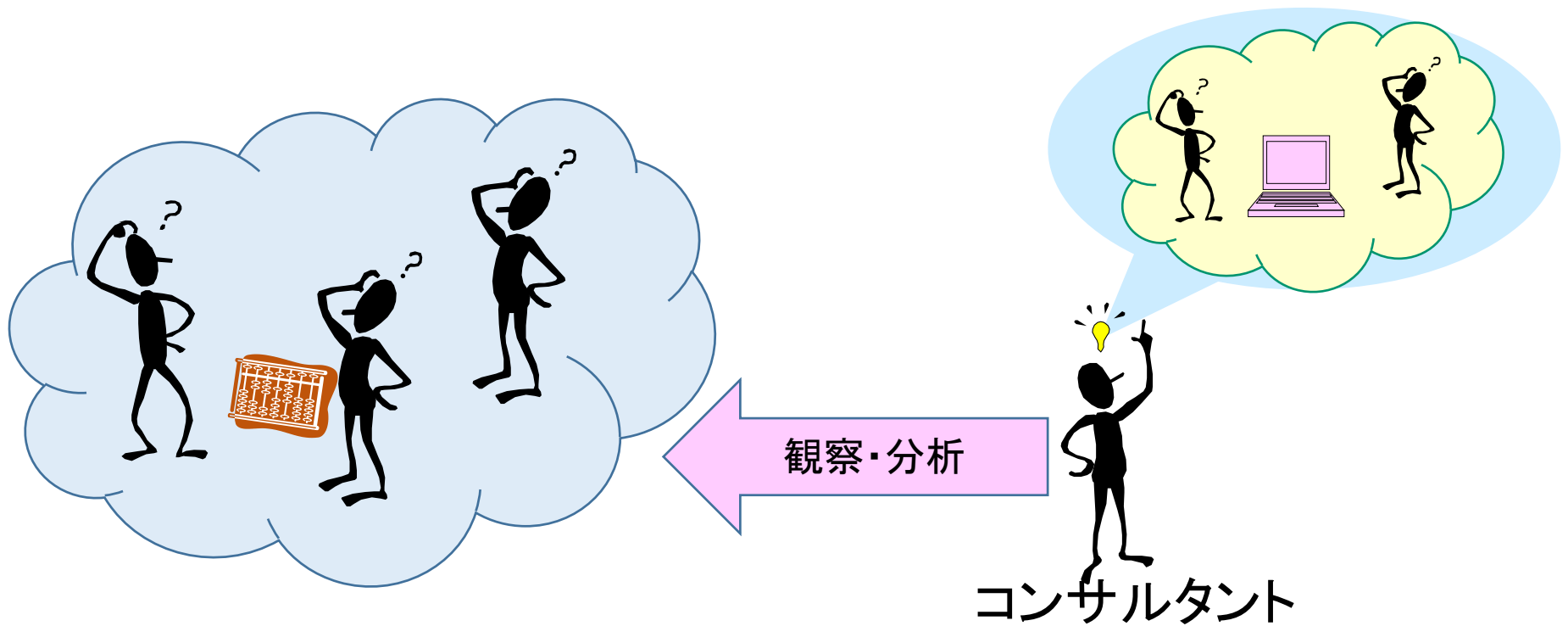
- 人々はなんらかの**仕事**や**活動**をしています。
 - 最初はソフトウェア、**コンピュータ**とは関係なく。
- 仕事例: **銀行業務**, **物品販売**, **ホテル経営**...
- 生活例: **音楽鑑賞**, **ゲーム**, **育児**, **介護**...

プログラミングより前って？ 2/4



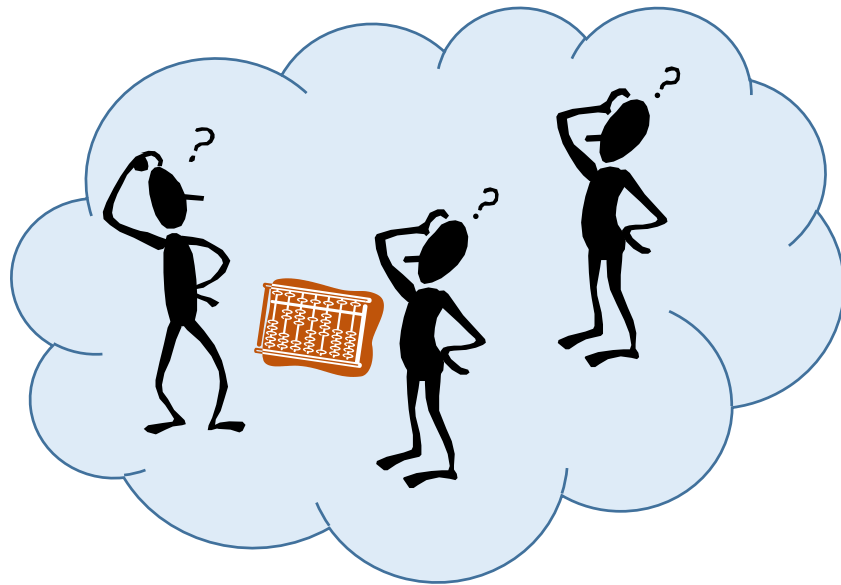
- コンサルタント等が仕事・生活を観察して...

モデリングとは？ 3/4



- ソフトウェアをどこかに導入したほうが仕事や生活がより良くなるかどうかをコンサルタント等が検討します。

モデリングとは？ 4/4



理解

観察・分析

コンサルタント



整理

開発指示



仕様書

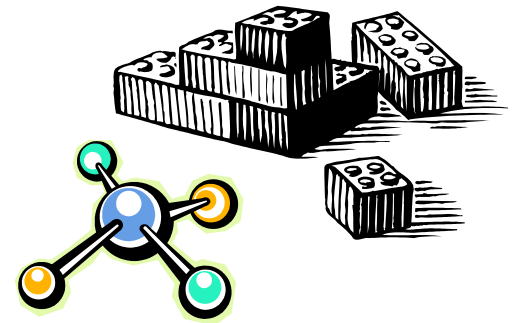
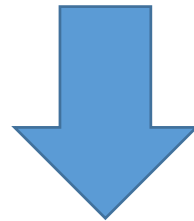


プログラマ

- **分析結果**に基づき、開発者や**プログラマ**等に、どんなソフトウェアを作ればよいか**指示**します。
- その後、プログラミング等が始まる...

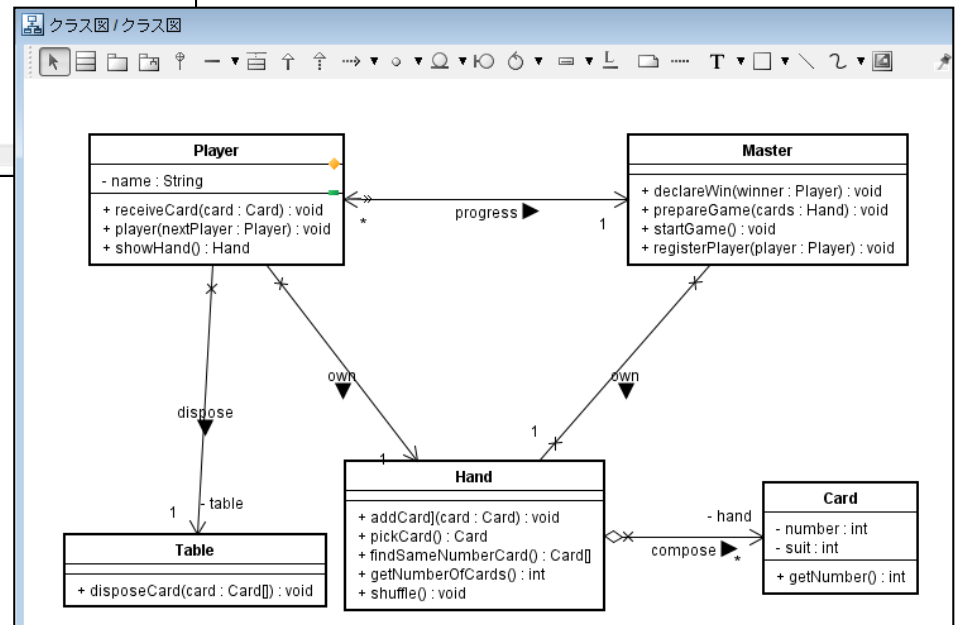
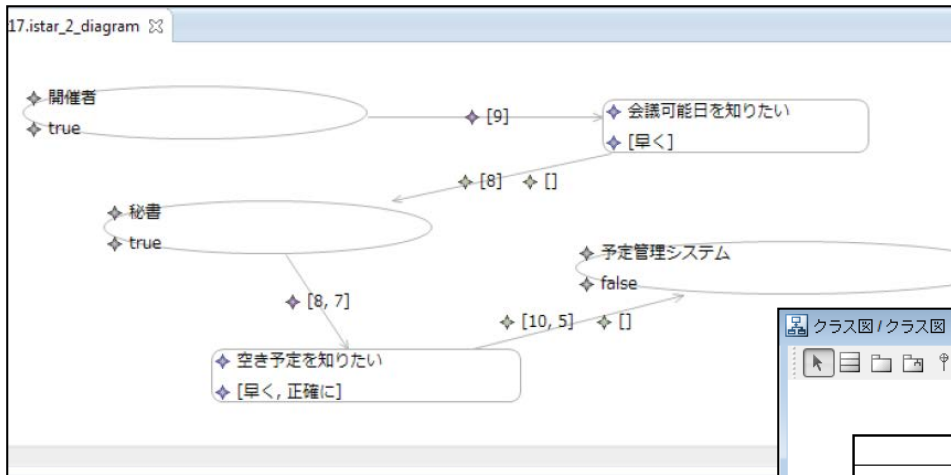
「理解」や「整理」をするためには？

- ソフトウェアの導入を検討している業務や生活の模型を作って,
- 模型の上で, 導入したらよさそうなソフトウェアを模擬的に導入し,
- 導入効果を吟味したりする.



- この模型のことを「**モデル**」, モデルを作ることを「**モデリング**」と呼ぶ.

モデルの例



ほとんどのモデルは上記のような**図式言語**で書かれる。

モデルの読者は誰？

- プログラマ等の開発者
 - モデルを「仕様」として用いる場合，当然，開発する人であるアーキテクト，ソフトウェア技術者，プログラマ等が読者となる。
- コンサルタント
 - モデルを書くのがコンサルタントだが，書いたモデルが正しいか随時吟味するため，当然，読者でもある。
- その他，利害関係者
 - 導入されるシステムを利用する人
 - システム導入で活動に影響(リストラとか)を受ける人
 - 導入で利益を得る人

要求と仕様の違い

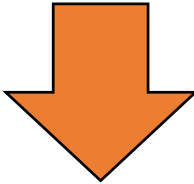
- 色々な人が色々な定義を提唱しているが、本講義では、M. Jackson の定義をベースにする。
- 要求 (Requirements)
 - 現実世界の事物がどうなってほしいかという人間の願望。
- 仕様 (Specification)
 - 要求を達成するために、現実世界に追加される情報システム(および付随する機械)の振る舞いの規定。
 - 仕様書の読者はプログラマ等の開発者である。
- 実務では要求と仕様は区別されないことがあるが、本講義では上記の定義を用いる。
- 尚、要求を「要件」と呼ぶ場合があるが、要件という用語は本講義では用いない。

要求と仕様の違いの例

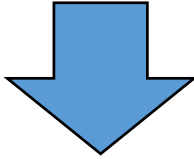


業務・活動	要求	仕様
動物園等の入場者を管理する業務.	入場券を買った人だけ入場できるようにしたい.	<ul style="list-style-type: none"> ● 入場券によってゲートが開く. ● 開いたゲートは一名のみ通過できる. ● 通過したらゲートは閉まる.
交差点の交通整理をする業務.	交差点で車等が衝突しないようにしたい.	<ul style="list-style-type: none"> ● 交差する道の信号機を互い違いに赤・青にする. ● 双方の道が青になる時間帯をつくらない. ● 双方の道が赤になる時間帯を作る.
集中治療室(ICU)での患者の容態監視.	患者の容態が急変したら, 看護師, 医師に緊急連絡したい.	<ul style="list-style-type: none"> ● 容態を示す情報(血圧等)を患者から常時得る. ● それぞれの情報の正常値を登録する. ● Etc...

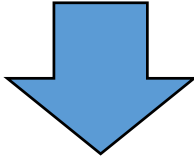
要求と仕様の関係

- 前頁の例からわかるように、要求を満たす仕様は複数存在しうる。(一個も存在しない場合もあるが)
 - それぞれの仕様は特定の理論や技法や機器を利用している.
 - 個々の理論や技法に注視する研究室が多いでしょう.
 - ある理論・技法・機器に過度に肩入れしていると、本来、それによって達成すべき要求を軽視してしまう場合がある.
- 
- 常に要求を念頭におき、別の解決法(仕様)を考えてみましょう.

本末転倒な例 1/2

- 以下の話は逸話で嘘ですが.
 - 要求「宇宙船で手書きのメモをしたい」
- 
- NASAの仕様
 - 「莫大な予算をかけて、重量が無くても書けるボールペンを開発！科学の勝利だ！」
 - 「莫大な予算をかけて、最新鋭の高信頼性タブレットを開発。科学の勝利だ！」
 - ソビエト(現ロシア)の仕様

本末転倒な例 2/2

- 以下の話は逸話で嘘ですが.
 - 要求「宇宙船で手書きのメモをしたい」
- 
- NASAの仕様
 - 「莫大な予算をかけて、重量が無くても書けるボールペンを開発！科学の勝利だ！」
 - 「莫大な予算をかけて、高機能タレット」
 - ソビエト(現ロシア)の仕様
 - 「エンピツと紙でいいじゃん。」

「要求仕様書」, 「RFP」

- 世の中には表題のような文書が多数ある.
- 要求仕様書
 - 前述の要求と仕様の双方が書かれたものが理想.
 - 実際は仕様が詳しく書かれており, 要求の説明が不十分なものが多い.
- RFP
 - Request for Proposal
 - 内容は要求仕様書とほぼ同じ. むしろ, 要求のみ書くのが理想.
 - 公募や入札等を通して, 実際に関与する企業体を募集するという意味合いが大きい.
 - 要求仕様書と同じく, 仕様は詳細であるが, 要求が不足している問題をはらむ.

開発の流れと関係者

① コンピュータに行なわせたいことを
理解する

誰が行なわせたいの？

② 理解したことを
説明できるレベルまで整理する

誰に説明するの？

③ コンピュータにわかる言葉に
翻訳する

誰が翻訳するの？

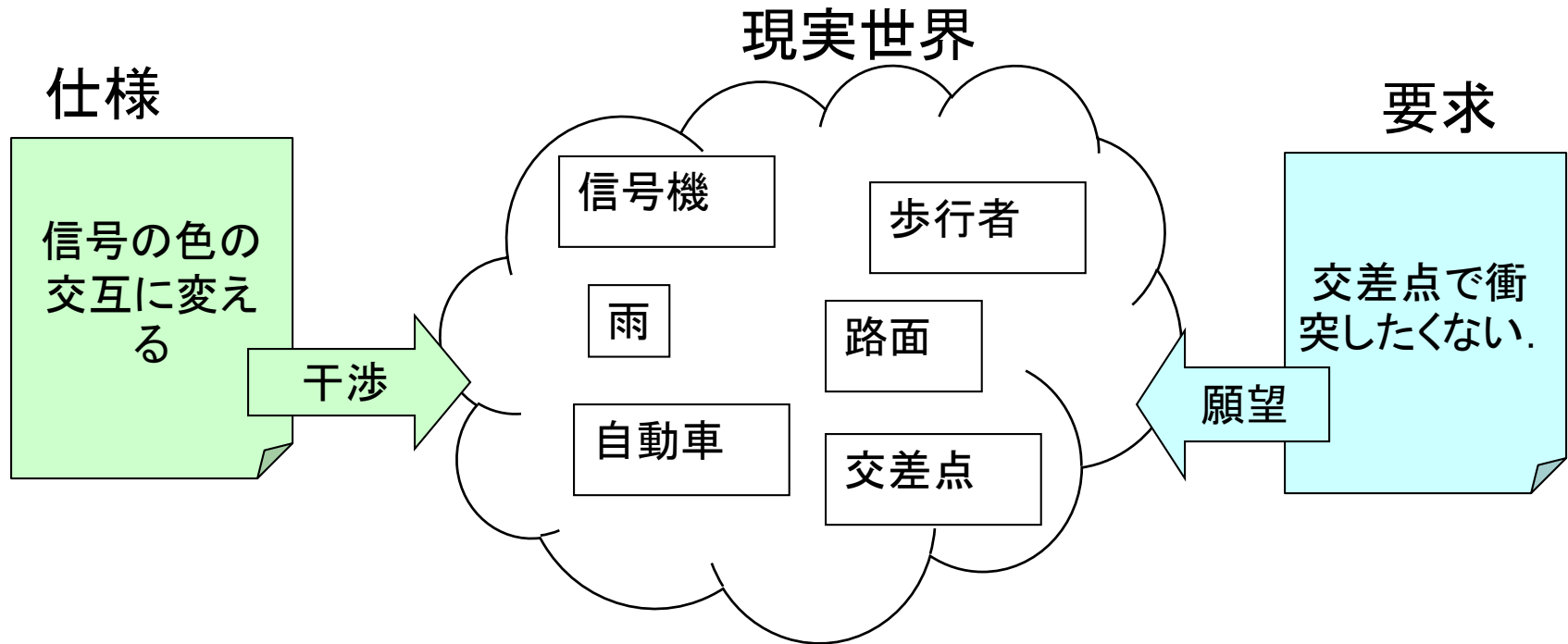
「行なわせたいこと」とは？

- 人間がコンピュータに行なわせたいことは、原則、**現実世界への干渉**である。
 - データを音の振動に変換して、人間に音楽を聞かせたい。
 - 電話番号リストを記憶させて、あとで電話番号を調べたい。
 - 交差点の信号機に色切り替えの命令を送り、自動車の衝突を避けたい。
 - 各自の思ってることを入力・共有することで、共感を得たり、情報収集をしたりしたい。
- **現実世界への干渉が無いコンピューティングは意味が無い！**

要求と仕様

- **要求**: 人間が**現実世界で、こうあってほしい、こう変化してほしい、こうなって欲しくない**と思ってること。
 - 例: 交差点で自動車が衝突して欲しくない.
 - 要件と呼ぶ人もいる. Requirement のこと.
- **仕様**: コンピュータが**現実世界に干渉する内容**や**行為を規定した文書**や**モデル**.
 - 例: 信号機に互い違いに赤, 青の信号を送る.
同時に青には絶対しない.
同時に赤の時間を少々とる.
 - コレを要求仕様と呼ぶ人もいる. Specification のこと.
 - 参考書にある「**コンピュータに行なわせたいこと**」

例: 信号制御の要求と仕様

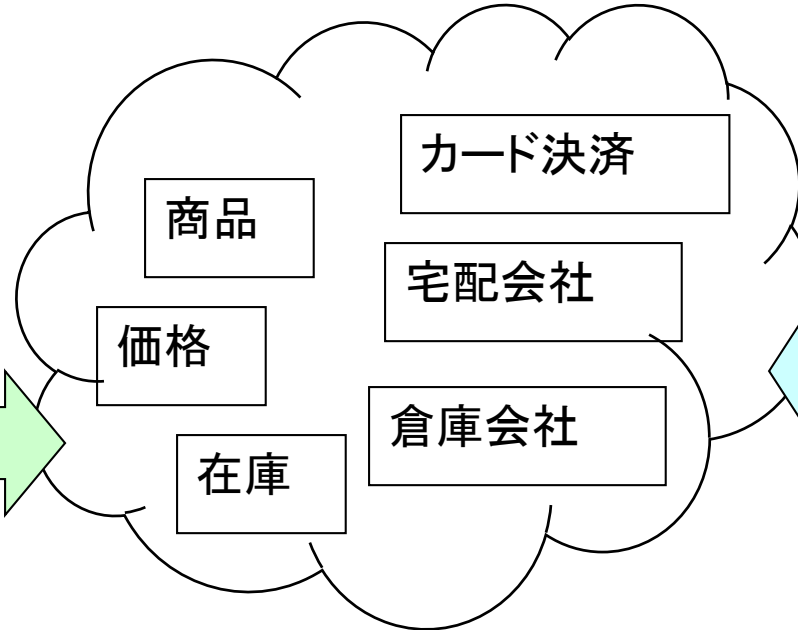


例: ネットショップの要求と仕様

仕様

- 商品リストをブラウザに表示.
- カード番号を受け取り, 決済する.
- 倉庫会社から宅配会社へ配送依頼.
- 在庫確認.

現実世界



要求

- 商品を選びたい.
- 支払いしたい.
- 品物を得たい.

例: MP3プレイヤーの要求と仕様

仕様

- 音楽配信サービスにアクセス.
- 再生履歴に基づき再生曲を決める.
- オートパワーオフ.

干渉

現実世界

電池

ネットワーク

再生履歴

配信サービス

ADコンバータ

耳

時計

スピーカー

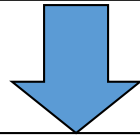
要求

- 音楽を聴きたい.
- 気に入った曲を聴きたい.
- 新しい曲を随時ききたい.
- いつでも聞きたい.

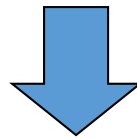
願望

ソフトウェア開発の流れ

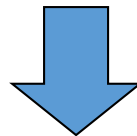
① 現実世界で起こって欲しい事を理解する



② コンピュータが行なえることを決定する



③ 決定したことを説明できるレベルまで整理する



④ コンピュータにわかる言葉に翻訳する

モデリングの代表的な側面

- **構造的側面**

- 現実世界のどんなモノが当面コンピュータで行いたいことに関係するか, それらの(静的な)関係は何かを明確にする.
- 通常, **クラス図**を利用.

- **機能的側面**

- 現実世界の事象(コンピュータへの入力)に対して, コンピュータは何を起こすかを明確にする.
- 通常, **ユースケースモデル**を利用.

- **振る舞いの側面**

- 機能の実行順序をモデル化.
- 通常, **ステートマシン図**, **アクティビティ図**, **シーケンス図**を利用.

ソフトウェアの整理法

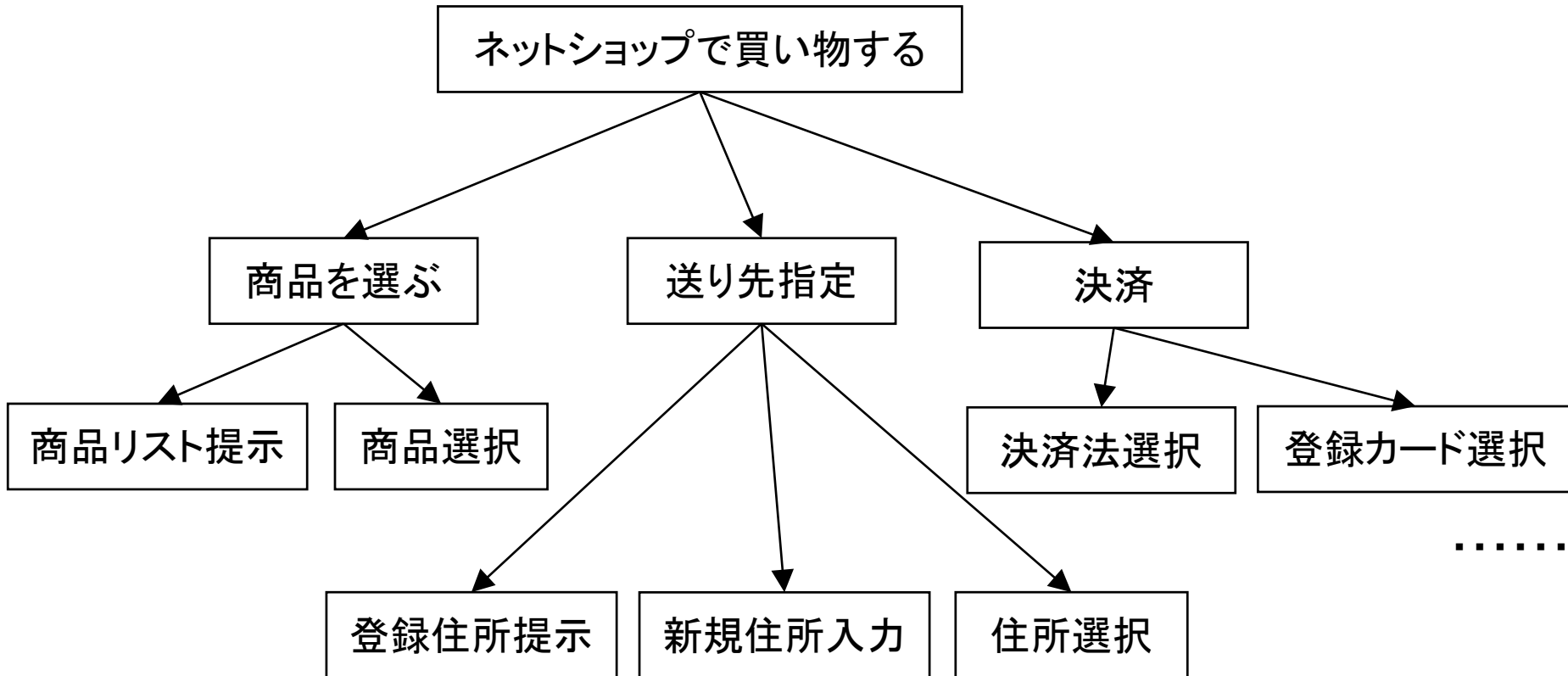
- オブジェクト指向
 - この授業のメインテーマ
- 構造化手法
 - 機能分解, 手続き型とも呼ばれる.
 - C言語等の従来型ソフトウェアの基本概念.
 - オブジェクト指向の中でも利用されている.

構造化手法とその問題点

- 基本理念
 - データ構造 + 処理(アルゴリズム) = ソフトウェア
- 機能(処理)をより簡単な機能群へ構造的に分解することで、ソフトウェアを整理する伝統的な手法.
- C言語やアセンブラ等, 歴史の古い言語の基本理念となっている.

- データと処理の関係付けが希薄.
- 上記の理由から, ソフトウェア中の部品(関数やデータ構造)をグループ化しても, グループの結束が弱い.

構造化の例



基本, このように,
複雑な機能を, より単純な機能に分解してゆく戦略.
C言語等は, この考え方に基づき作られている.

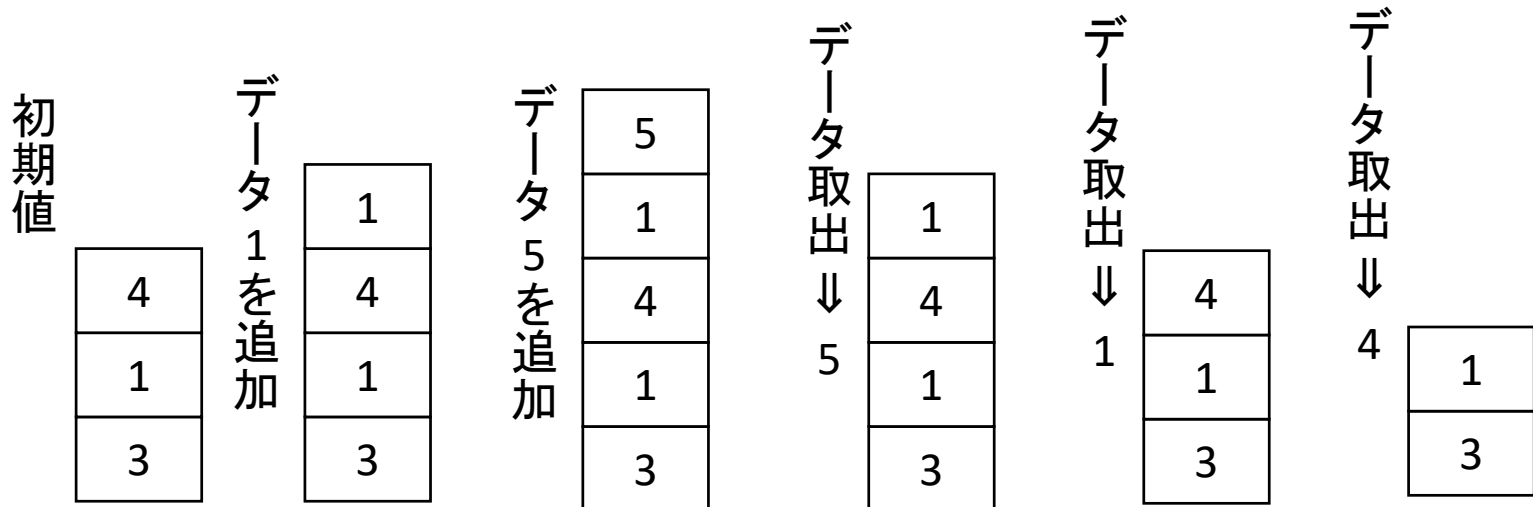
オブジェクト指向

基本的に以下の二つのアイデアの融合

- **抽象データ型** (Abstract Data Type, **ADT**)
 - データを直接的なデータ格納表現では無く,
 - データに適用できる関数(群)として定義する考え方.
 - **利点:** データ表現という実現方法に縛られずに, データとは何かを定義できる.
- **汎化, 継承**に基づく**差分記述**
 - ADTは関数の集合であるが,
 - その集合に新しい関数を追加したり,
 - 既存の関数の処理内容を更新したりすることで,
 - 新しいデータ型を定義する考え方.
 - **利点:** 新しい定義をする際に差分のみ書けばよいので楽.

ADTの例 1/2

- スタックというデータ構造はご存知だと思う。
- 下の図のように、後から入れたデータが先に取り出せるという、一風変わったデータ構造である。
- コンパイラ等の字句解析にとっても便利な構造として知られている。
- 一般的な説明は下図を使う場合が多い。



ADTの例 2/2

- 前頁のように具体的な箱のイメージでは無く,
- スタックというものを以下の**二つの関数で操作できる「何か」**として定義する.
 - 関数の名前はなんでもいいのだが.

push(x)

- データxを追加する.

pop()

- 直近に追加したデータを得られる.
- そのデータはスタックから削除される.
- データが空の場合, 何もしない.

- 「どのように(how)」は語らず, 「**何が起こるか(what)**」のみを指定するため, 実装の時の自由度が増す.

継承の例

- 前述のスタックは保持できるデータの上限が無い.
- これに対して, 例えば「**上限10個のスタック**」という**特化したスタック**を作る場合, pushのみ更新すればよい.

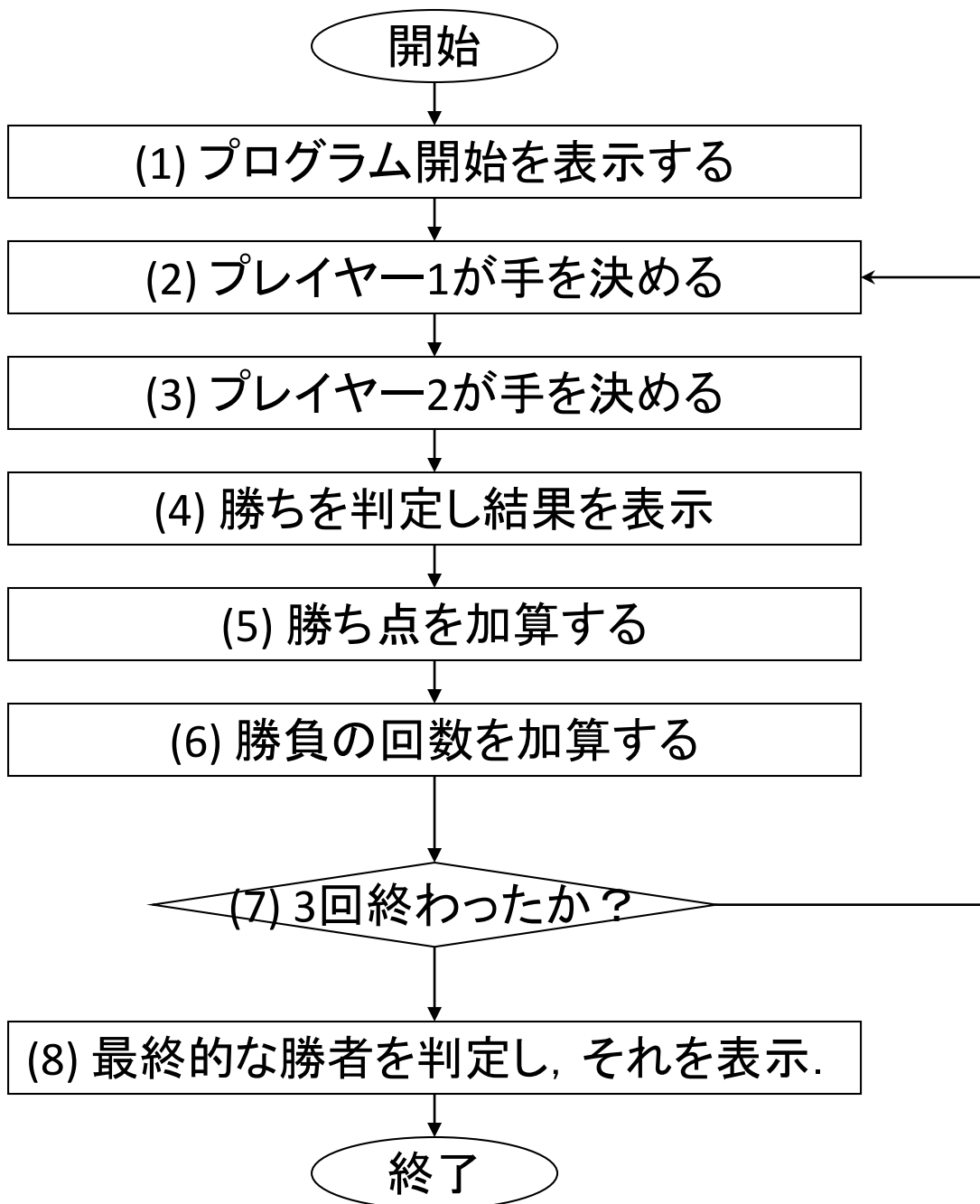
push(x)

- データ数が9個以下の場合, データxを追加する.
 - データ数が10個以上の場合, 何もしない.
-
- pop()のほうは, そのままでよく, 修正はおよそ半分で済む.

Java即オブジェクト指向では無い

- oo02sample.zipにJavaで書いた, オブジェクト指向ではないプログラムを挙げる.
 - 注意: **オブジェクト指向としては最悪なプログラム!**
- これを用いて, **Java即オブジェクト指向では無い**ことを説明する.
- SimpleJanken.java
 - じゃんけんの振る舞いを模倣するプログラム
- SimpleMath.java
 - 簡単な数値計算, 整数の平方根と最大公約数.
- オブジェクト指向らしいJavaのプログラムは次週.

じゃんけんの アルゴリズム



整数の平方根を求める問題

- 問題: 非負整数の平方根を求めよ. ただし, 解が整数とならない場合は小数点以下を切り捨てよ.
- 理論: 以下の数学的事実が知られている.

$$(2 \times 1 - 1)$$

$$(2 \times 2 - 1)$$

$$1 + 3 + 5 + \dots + (2n - 1) = n^2$$

足し合わせている項の数が平方根
(この例ではn個)