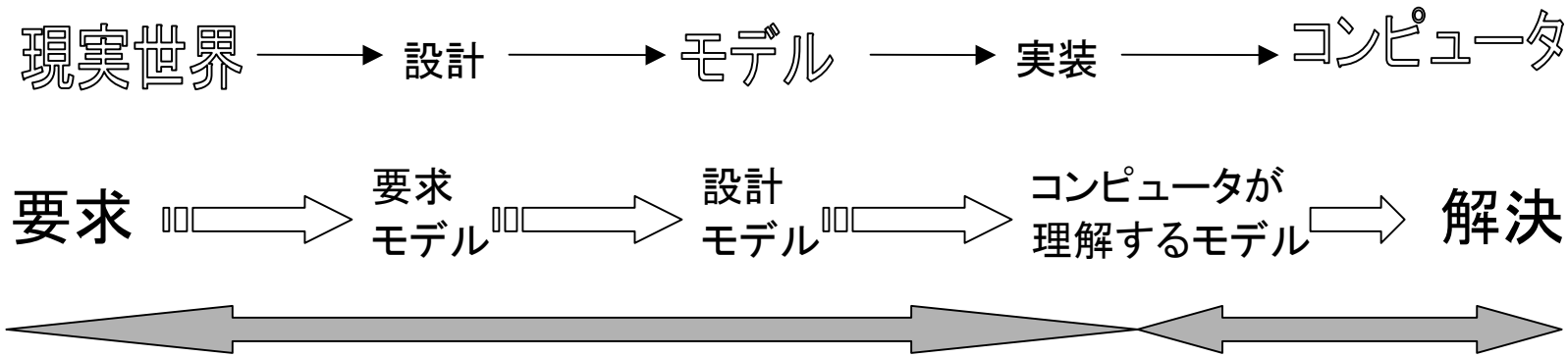
パラダイム三題噺　Part-1

オブジェクト指向

日立ソフトウエアエンジニアリング（株）
教育センタ部
久保　隆

現実世界 と モデル と コンピュータ



パラダイム

プロセスフロー
データフロー
リレーション
ディシジョンテーブル
スプレッドシート
状態遷移

コンピュータ言語

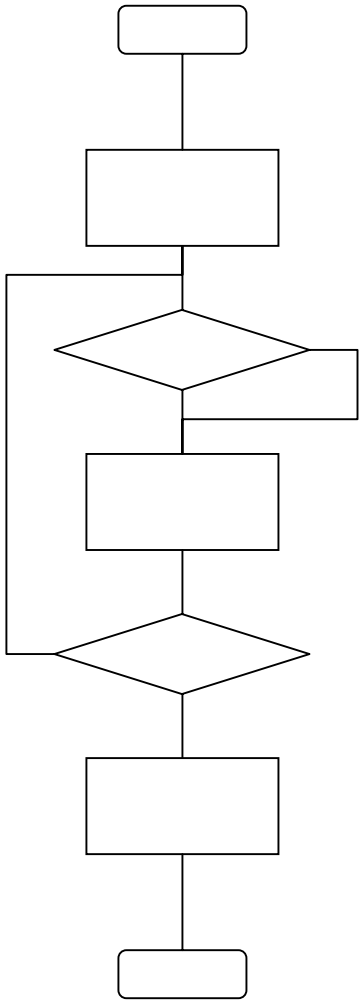
COBOL

C

RDB

EXCEL

プロセスフローモデルの場合



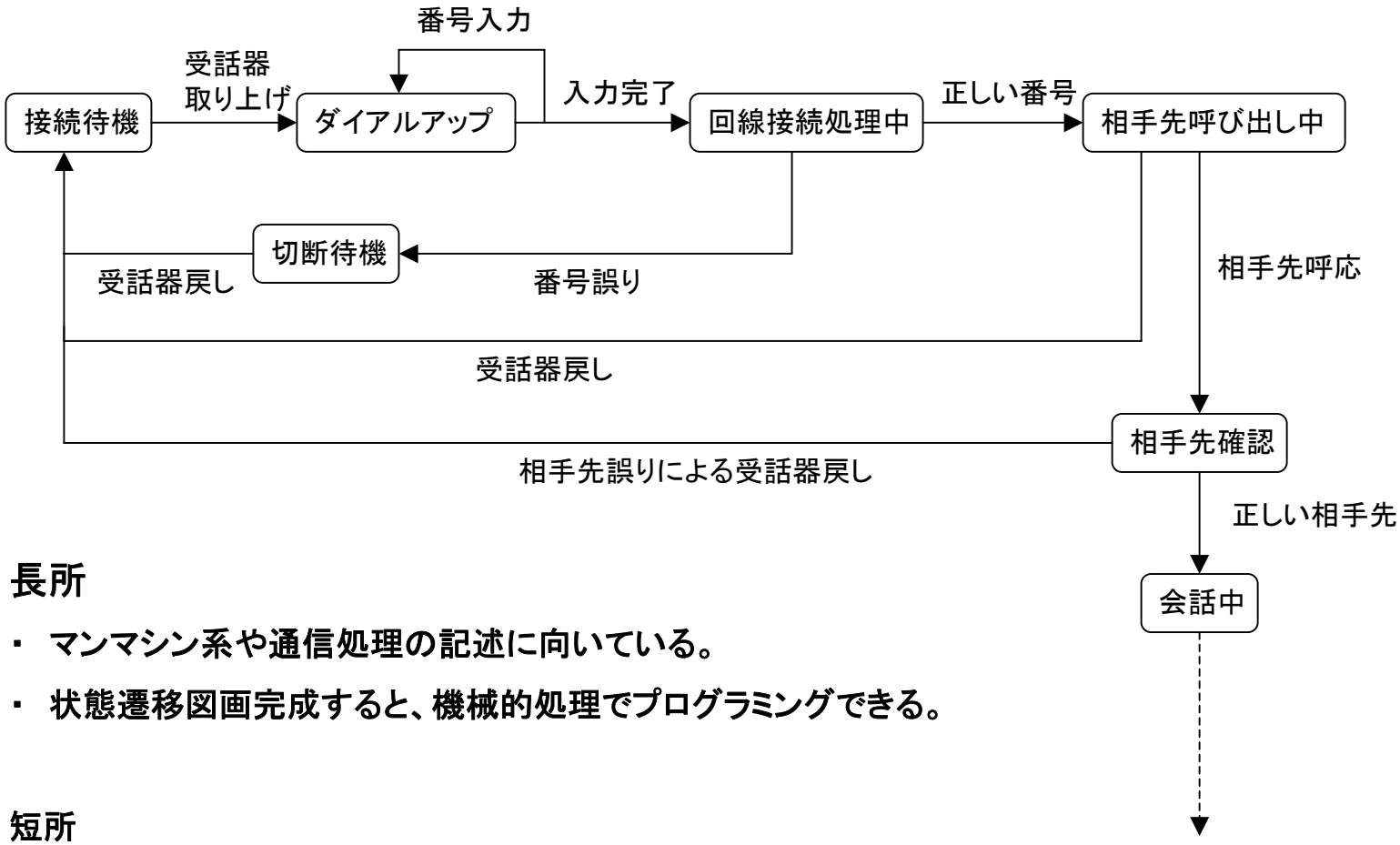
長所

- 現在使われているコンピュータの処理手順と相性がよい。
  したがってマシン語への変換が比較的容易
- 数学の解法や、料理のレシピなど、さまざまな分野で「手順」
  として使われており、人間との相性がよい。

短所

- 記述するシステムの規模が大きくなると、全体が見通せない。
- 複数のモデルの間での類似や相違がわかりにくい。
  したがって部品化が困難
- 何が入力で何が出力なのかが明確ではない。
  書き手のスキルに大きく依存する。
- 併行処理がうまく記述できない

状態遷移モデルの場合



長所

- **マンマシン系や通信処理の記述に向いている。**
- **状態遷移図画完成すると、機械的処理でプログラミングできる。**

短所

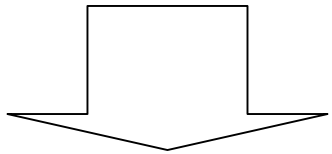
- **状態が整理できないシステムでは、まったく役に立たない。**
- **入力や出力に依存する処理が明確に記述できない。**

旧来のモデルの欠陥

- 現実世界とモデルとの間に乖離がある
- 適応できる分野が限られる
- パラダイムに一貫性が無い
- 不統一なモジュラリティにより再利用が困難

オブジェクトモデル

そもそもの始まりは
XEROX社　PARC　での　Alan　Key　等による　実験プロジェクト
DynaBook　用のコンピュータ言語　**SmallTalk**
子供から老人まで使えるシステムを目指した結果、
現実世界そのものをパラダイムとして**オブジェクト指向**が開発された。
SmallTalk－80（1980）　でほぼ完成
その後のオブジェクト指向の考え方は、全てこれを規範としている。

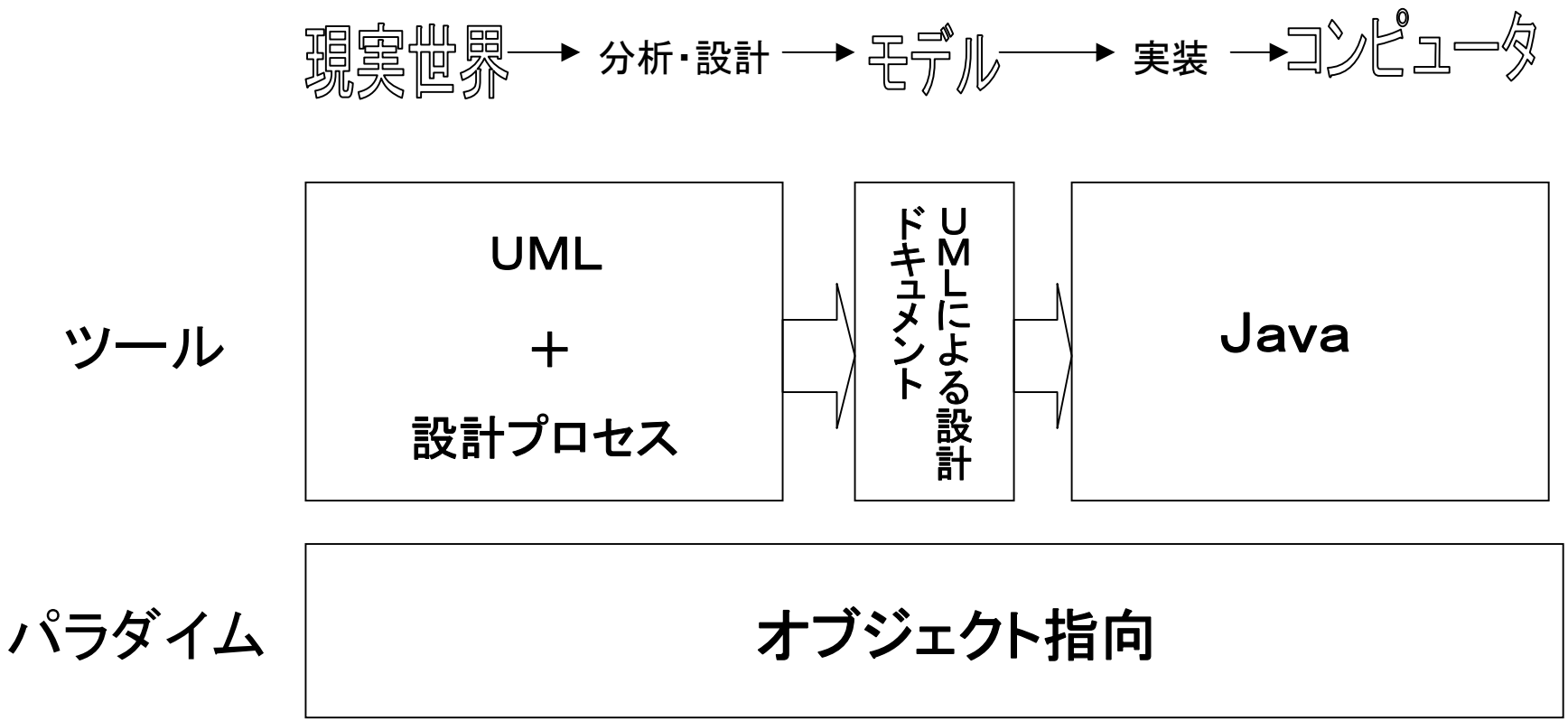


オブジェクト指向はプログラム言語でまず最初に開発された

・バックに厳密な規範（プログラム言語）がある
・設計の収斂の方向が明確である

（オブジェクト指向の理解には、プログラム言語からのアプローチが早道）

オブジェクト指向・Java・UMLの関係



オブジェクトモデルのパラダイム

現実世界を極力そのままでモデル化する

静的構造

現実に存在する物（オブジェクト）を、「データ」と「振舞い」とに整理し、
現実世界のモデルを作り上げる。

動的構造

物（オブジェクト）同志のメッセージのやり取りにより
物事が処理される（問題が解決される）。

オブジェクトとは（静的構造）その１

現実世界に存在するものは、全てオブジェクトになり得る
抽象的存在であってもかまわない
ただし、オブジェクトは他のオブジェクトと区別できなければならない

[例]

- A男さんの持っているTV
- B子さんの持っているTV
- A男さんの５万円の支払い
- B子さんの５万円の支払い
- X販売店が受け取ったビール５ダースの注文
- A男さんのB子さんに対する愛情
- A男さんとB子さんの、伊豆の温泉への一泊旅行

オブジェクトとは（静的構造）その２

それぞれに固有のデータを持っている

[例]

久保の普通預金口座

（属性）	（値）
・支店番号	1234
・口座番号	56789012
・預金残高	5万円
・自動引き落とし項目	電気代 ガス代 電話代

山田の普通預金口座

（属性）	（値）
・支店番号	4321
・口座番号	12345678
・預金残高	100万円
・自動引き落とし項目	電気代 水道代 NHK受信料 クレジットカード

オブジェクトとは（静的構造）その３

それぞれに固有の振る舞い（メソッド）を持っている
外部からの刺激（メッセージ）に対して何らかの反応する仕組みを持っている
自らのデータを、どのように操作するかを知っている

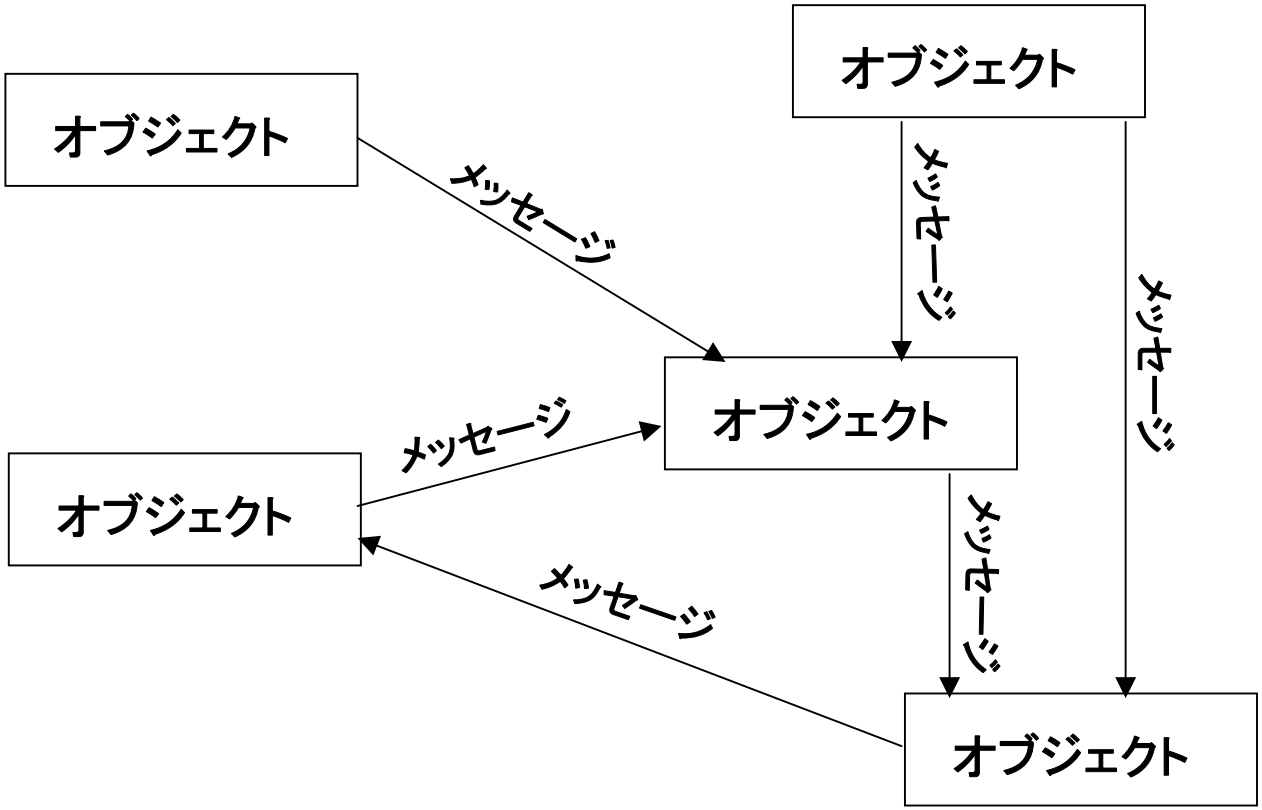
［例］

・アクセルを踏むと車は加速する（速度のデータを増加させる）
・「お手」と言うとポチは手を出す（前足の状態を、差し出した状態にする）

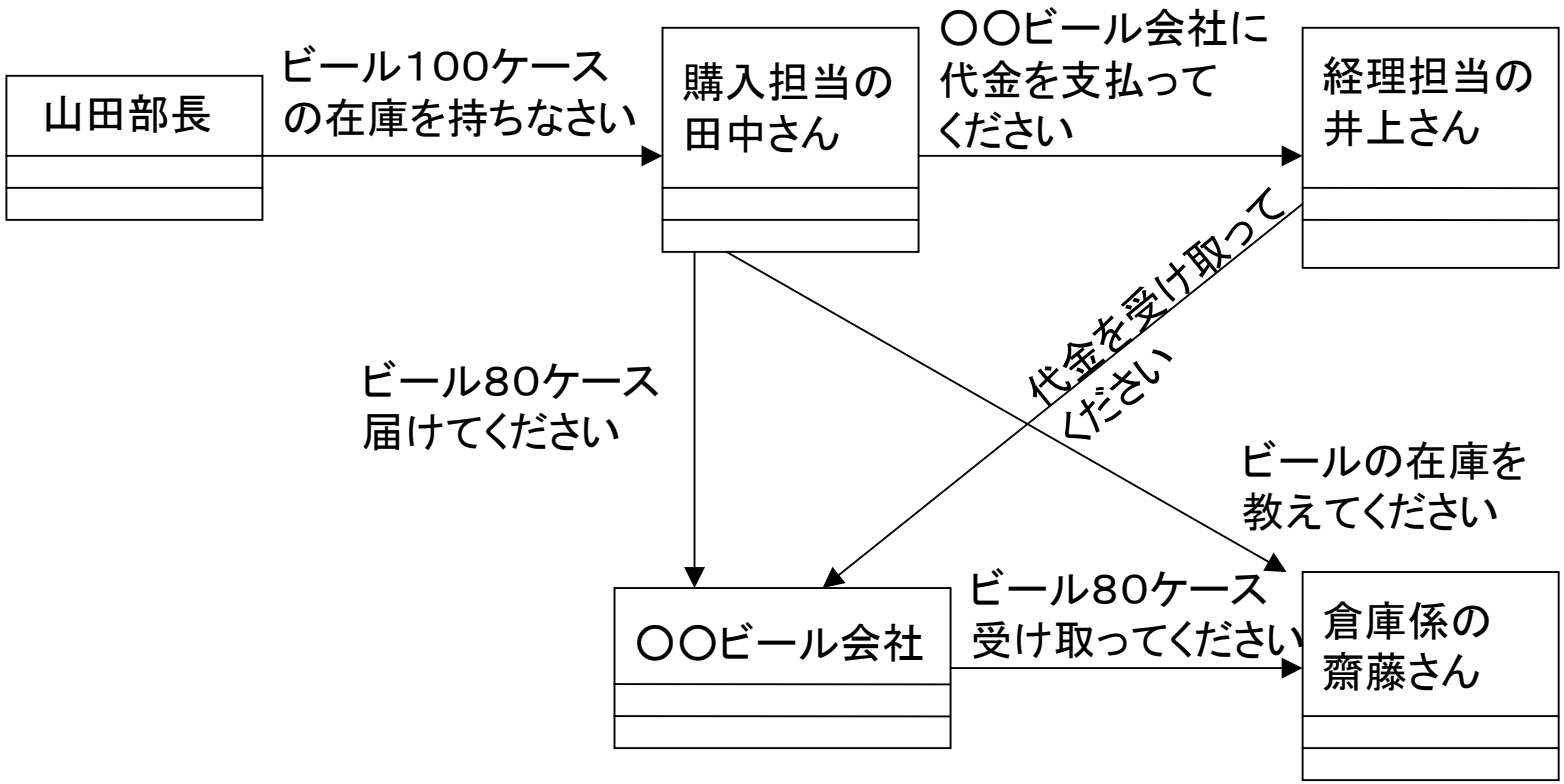
久保の普通預金口座
・３万円預金すると、預金残高は８万円になる
・５万円引き出すと、預金残高は３万円になる

山田さんの普通預金口座
・３万円預金すると、預金残高は１０３万円になる
・１１０万円引き出すと、総合口座で定期預金に２０万円あるので、
これを担保とし１０万円のローンが発生する

メッセージとは（動的構造）その１

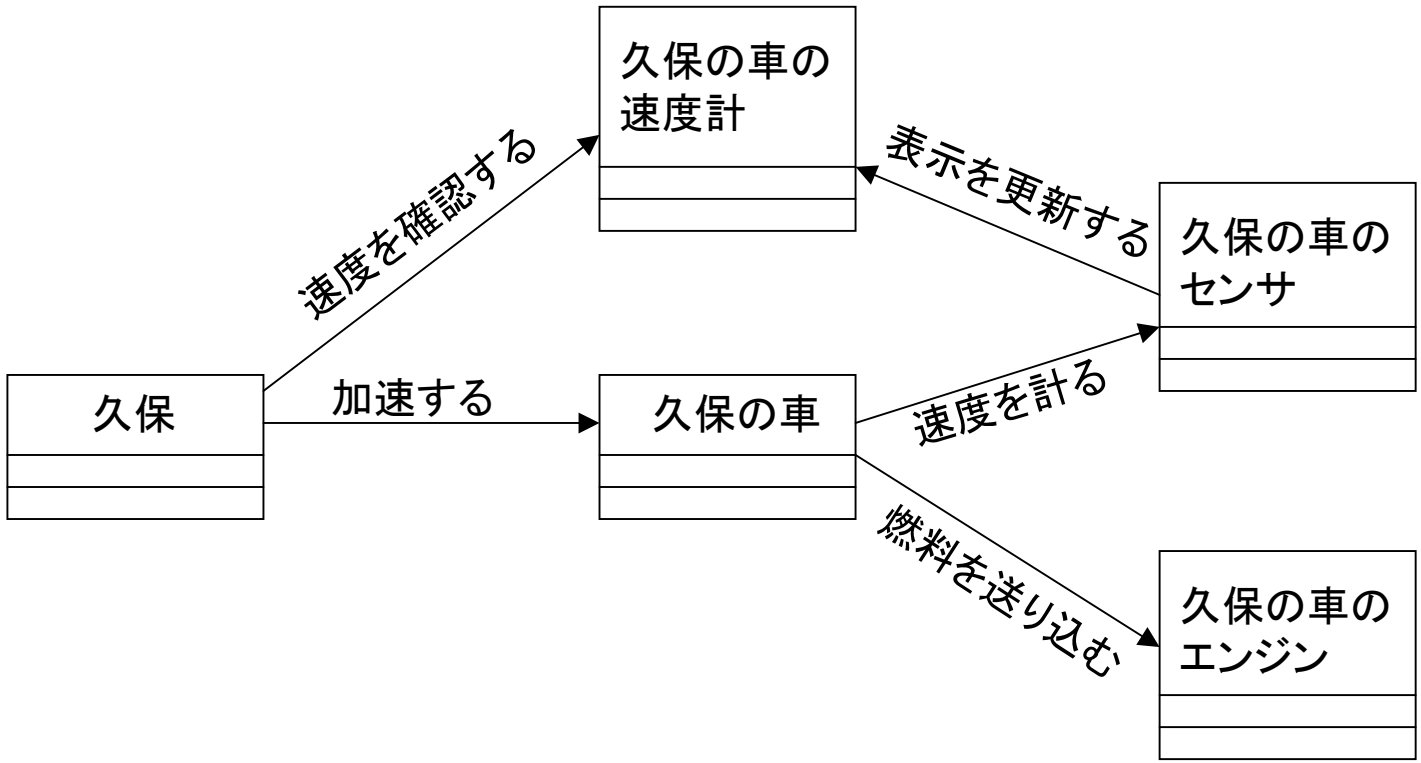


メッセージとは（動的構造）その２



他のオブジェクトにメッセージを送る機能を**メッセージ・センディング**と呼ぶ

メッセージとは（動的構造）その３



他のオブジェクトにメッセージを送る機能を**メッセージ・センディング**と呼ぶ

データと振舞いの関係　その１

オブジェクト指向モデル

データ自身が自分の値を
どのように変更するか、
その手続き（振舞い）を
知っている。

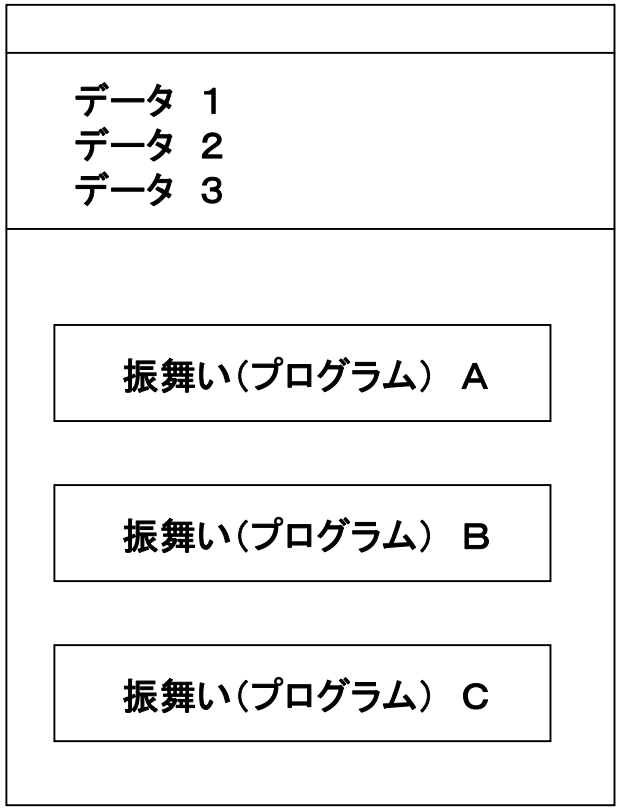
旧来のモデル

データと手続きは独立
データの変更は外部の
手続き（プログラム）により
行う。

データと振舞いの関係　その２

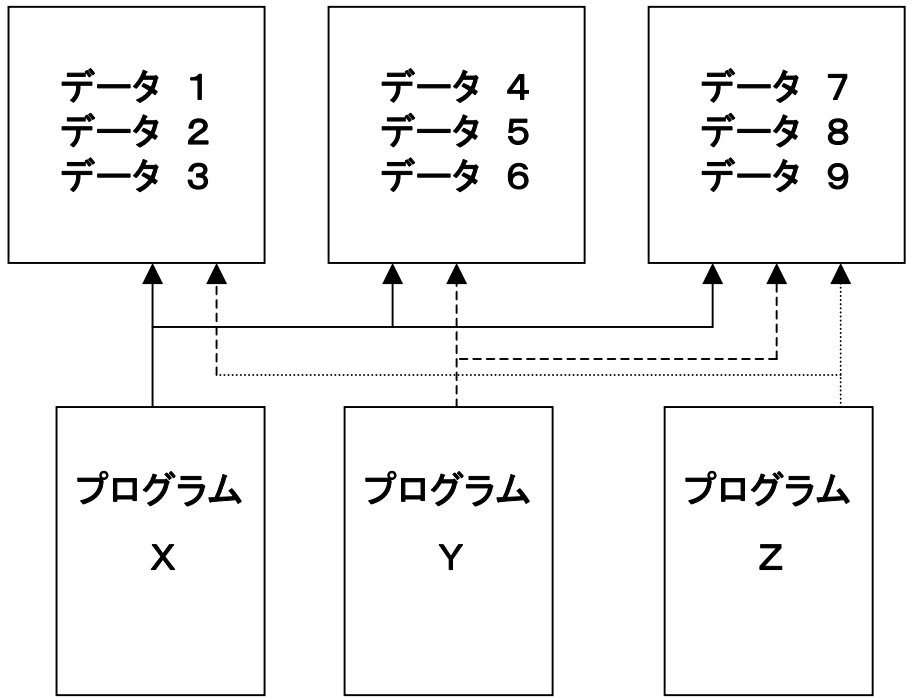
オブジェクト指向

オブジェクト



従来のパラダイム

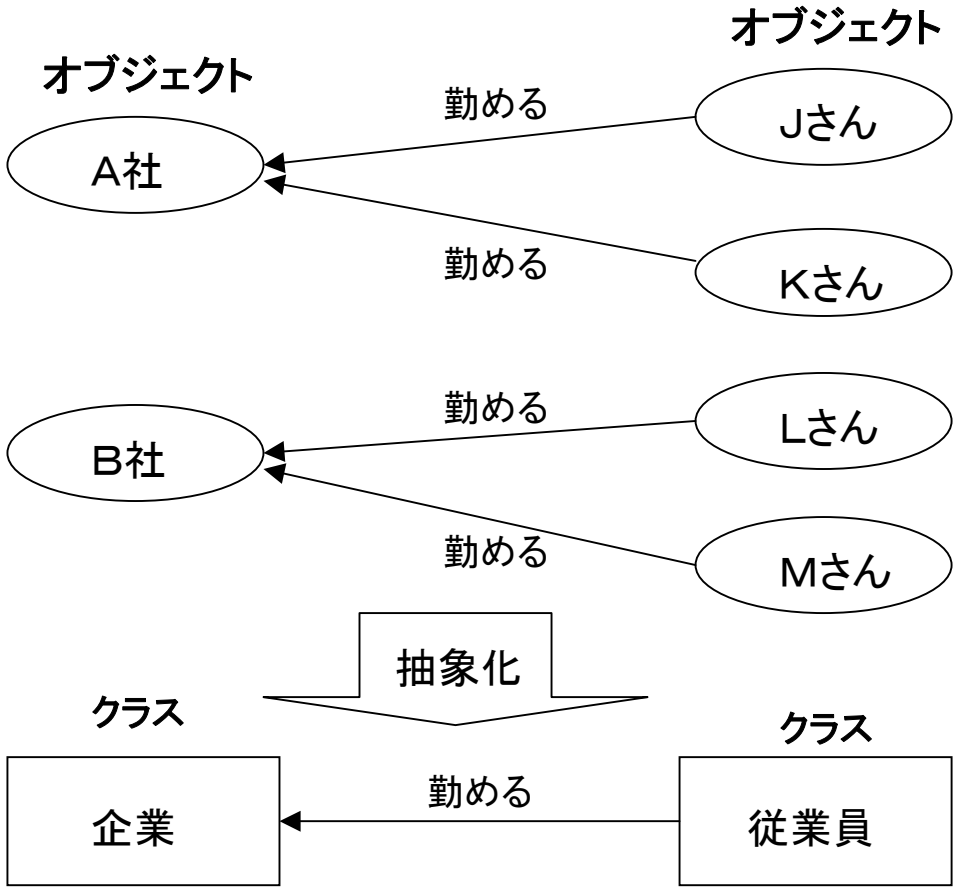
レコード



データ操作プログラム

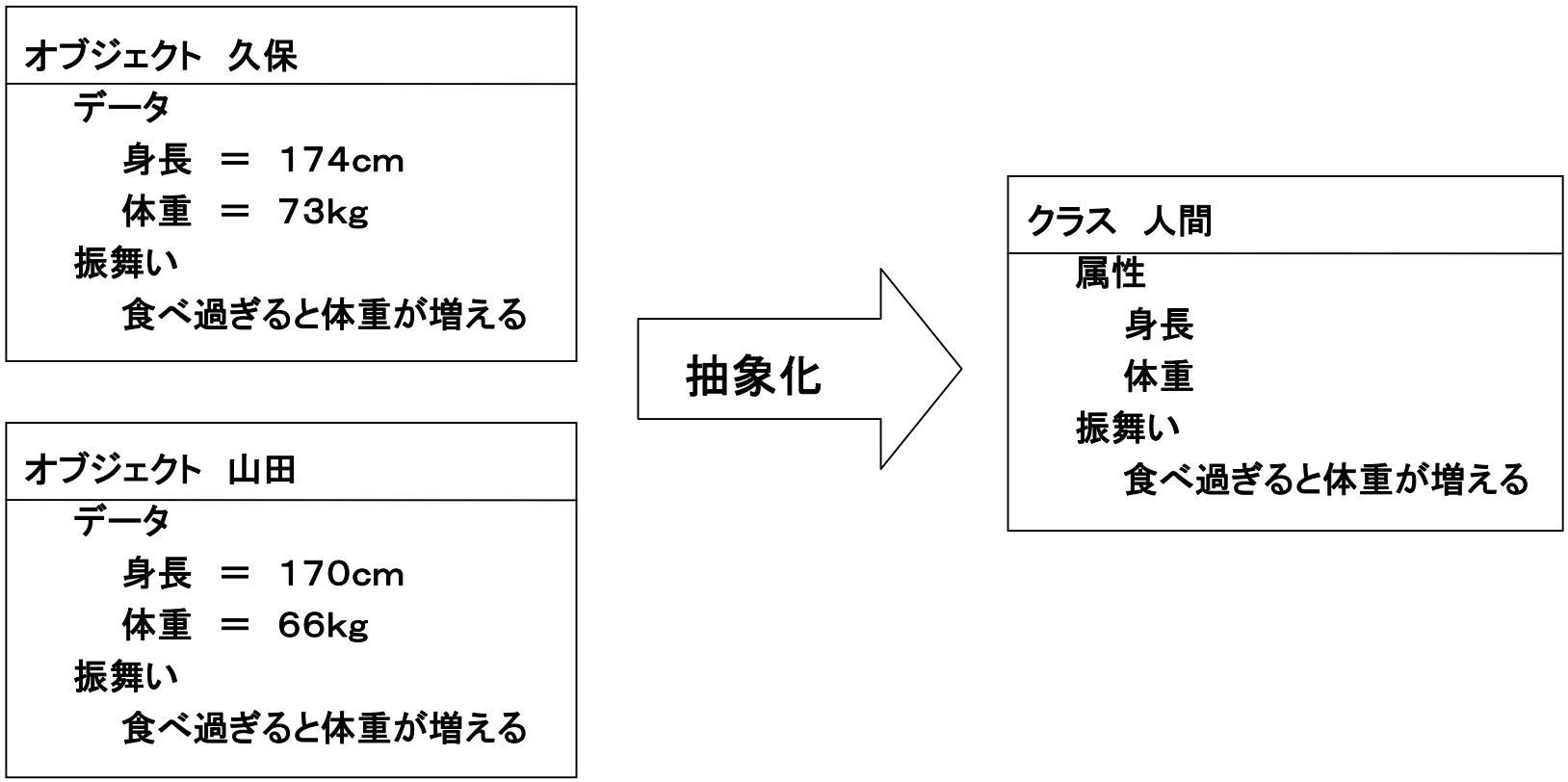
オブジェクトとクラス

オブジェクトを抽象化したものがクラス



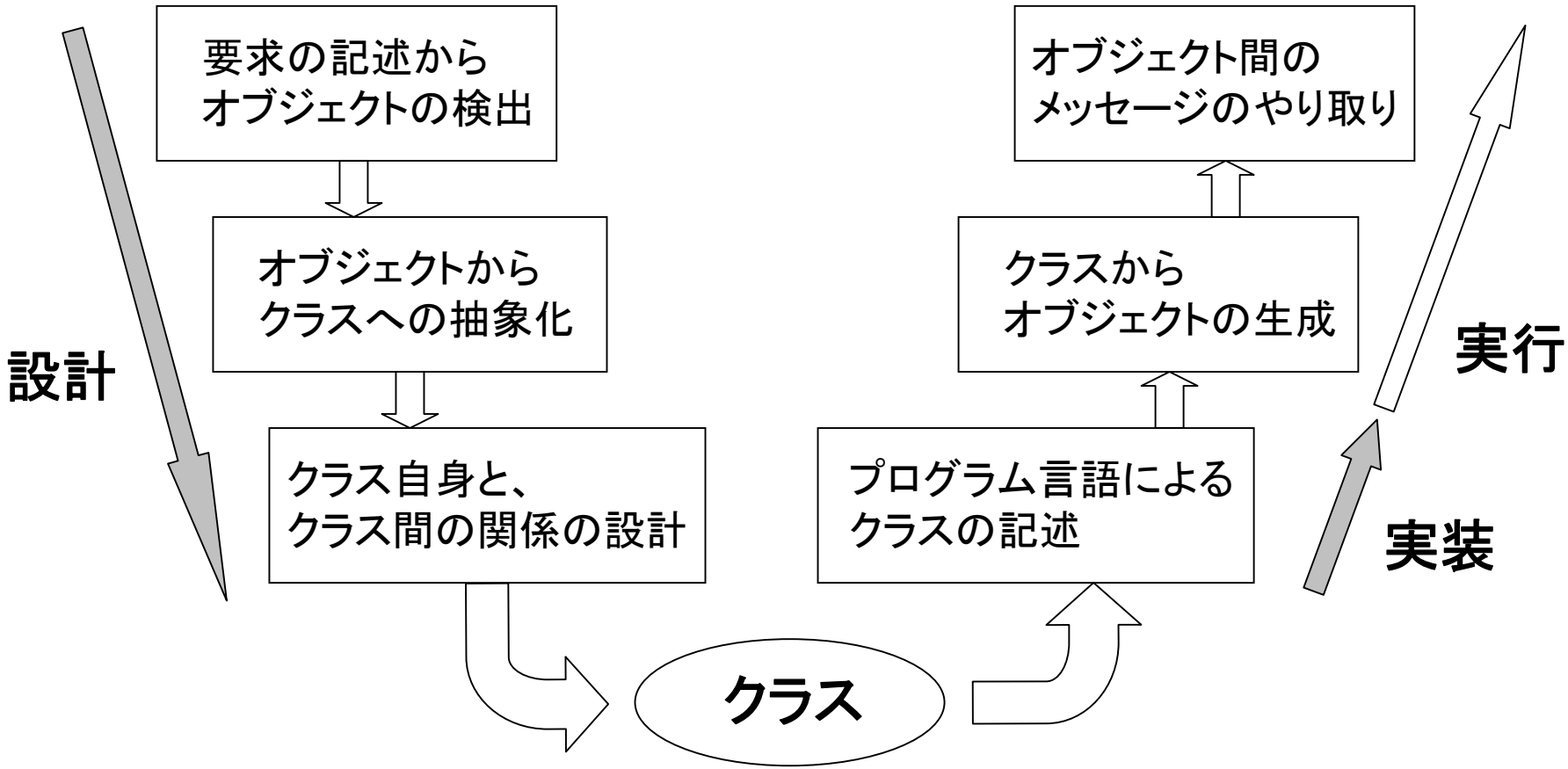
データから属性へ

オブジェクトのクラスへの抽象化の過程で、
- ・オブジェクトのデータは属性に抽象化される。
- ・オブジェクトの振舞いはそのまま集約される



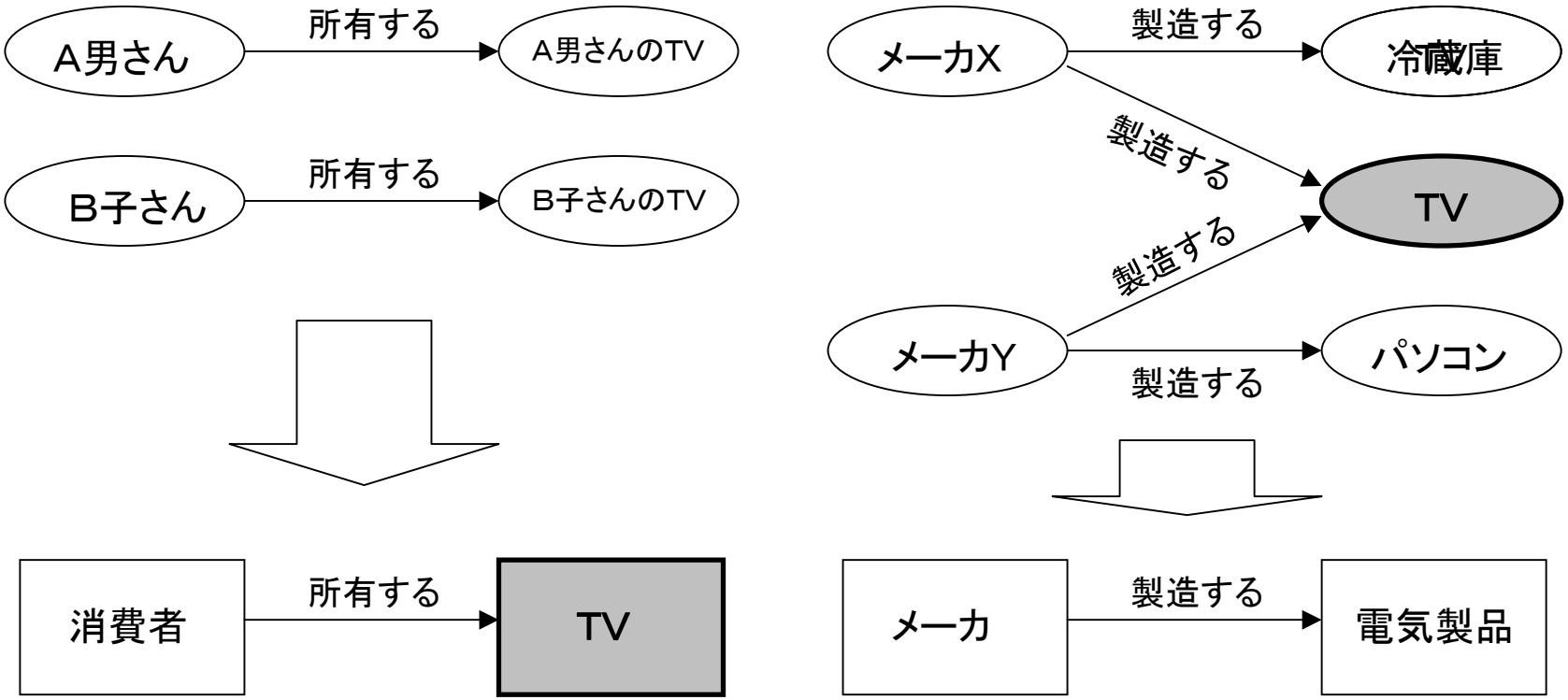
オブジェクトとクラスの使い分け

一般的には、個別のオブジェクト一つひとつについてデータや振る舞を
考えるのは面倒なので、設計や実装を考える場合、クラスに抽象化して進める。



オブジェクトからクラスの抽出

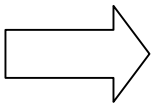
何をクラスとするかはアプリケーションの目的によって異なる

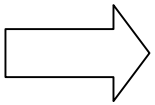


クラスの中身

属性

振る舞い
（メソッド）





人　クラス
生年月日 氏名 住所 機嫌 好きな人 嫌いな人
歳はいくつ？ 算数の問題を解く 物を運ぶ 好きな人に出会うと機嫌がよくなる ご機嫌いかが？

オブジェクトの中身（その1）

値の入った属性 ⇨

実行できる振る舞い（メソッド） ⇨

人　クラスの　久保　オブジェクト

生年月日	⇨	昭和18年11月20日
氏名	⇨	久保　隆
住所	⇨	鎌倉市
機嫌	⇨	不明
好きな人	⇨	A子さん、B子さん
嫌いな人	⇨	J男君、K男君

歳はいくつ？

算数の問題を解く

物を運ぶ

好きな人に出会うと機嫌がよくなる

ご機嫌いかが？

オブジェクトの中身（その2）

値の入った属性 ⇨

実行できる振る舞い（メソッド） ⇨

人　クラスの　山田　オブジェクト

生年月日	⇨	昭和25年3月15日
氏名	⇨	山田　太郎
住所	⇨	横浜市
機嫌	⇨	よい
好きな人	⇨	H子さん、J男君
嫌いな人	⇨	A子さん、K男君

歳はいくつ？

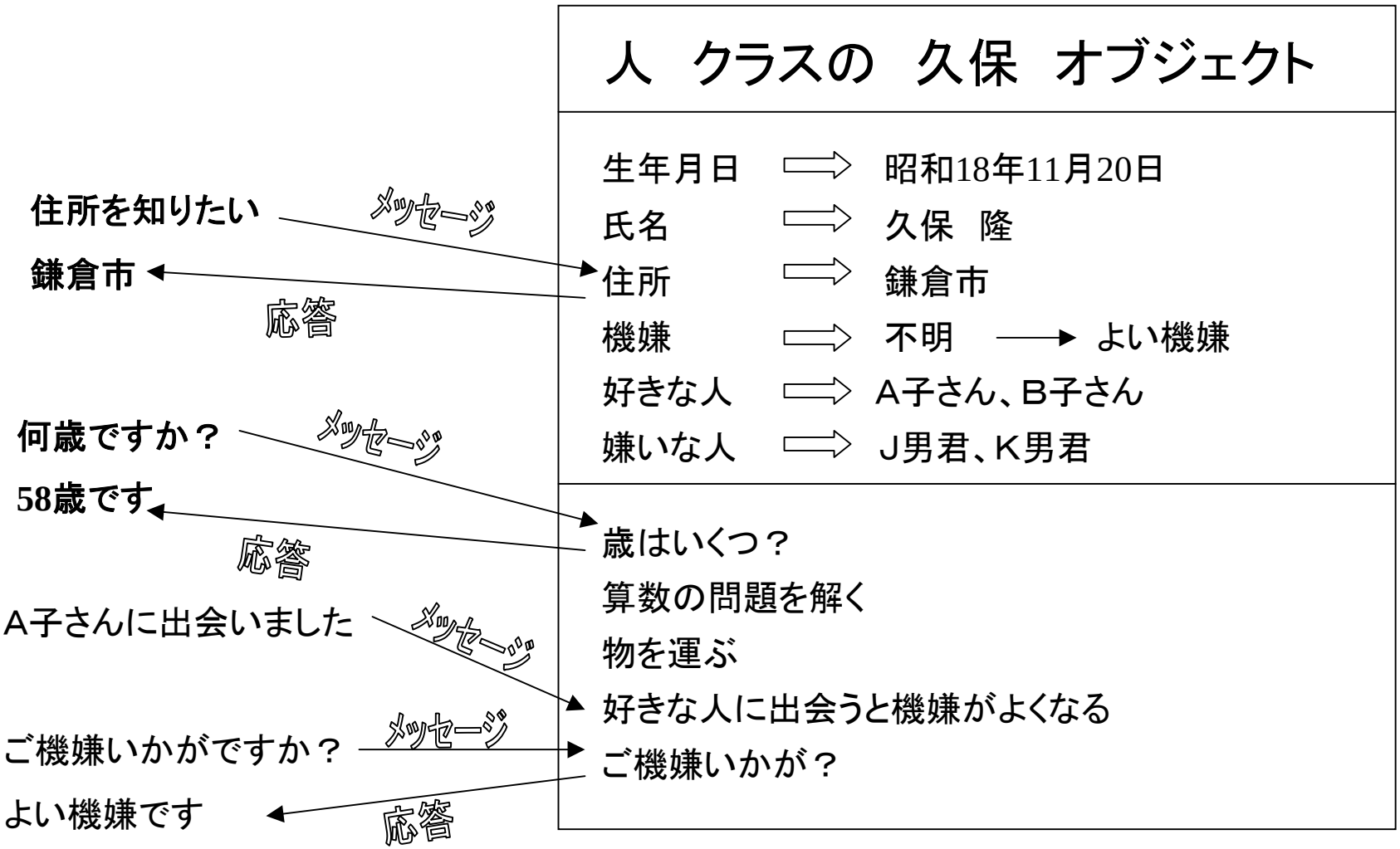
算数の問題を解く

物を運ぶ

好きな人に出会うと機嫌がよくなる

ご機嫌いかが？

オブジェクトがメッセージを受け取ると



普通預金口座の場合

属性 ⇒

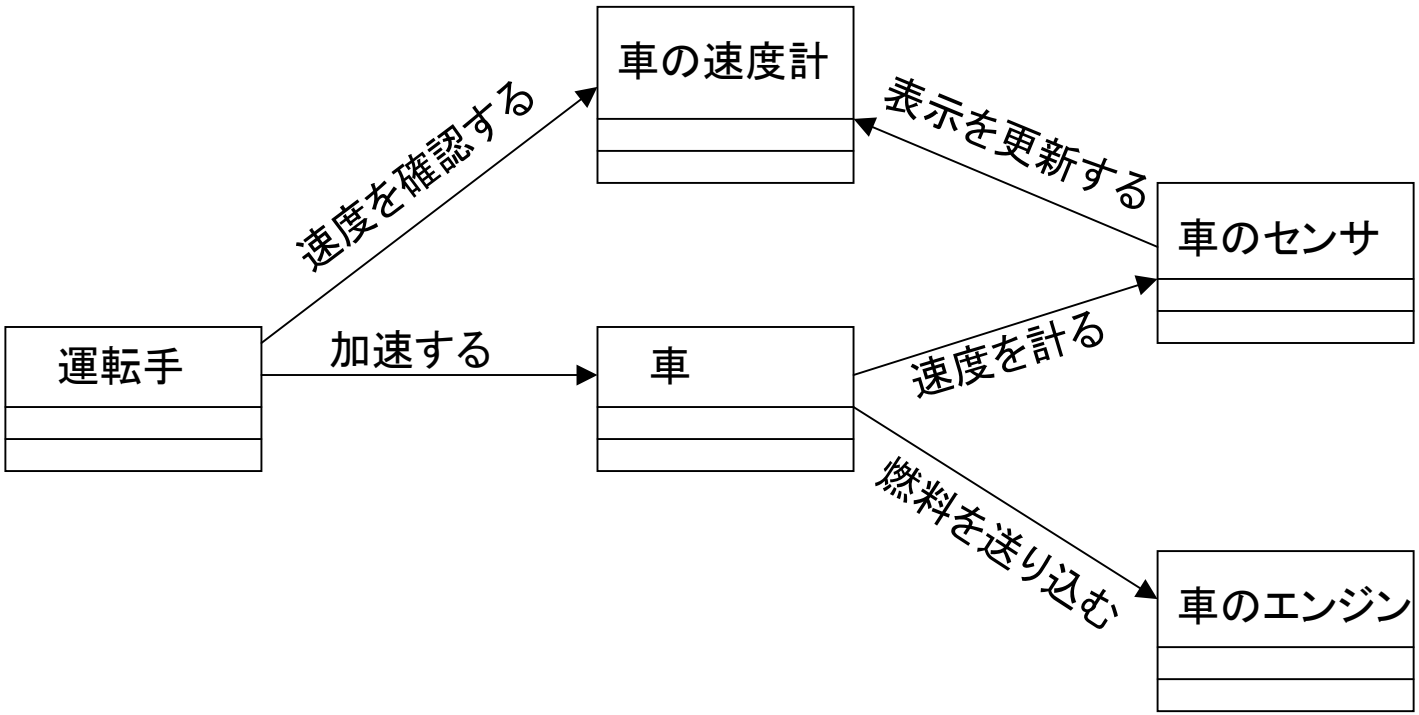
振る舞い ⇒

普通預金口座　クラス

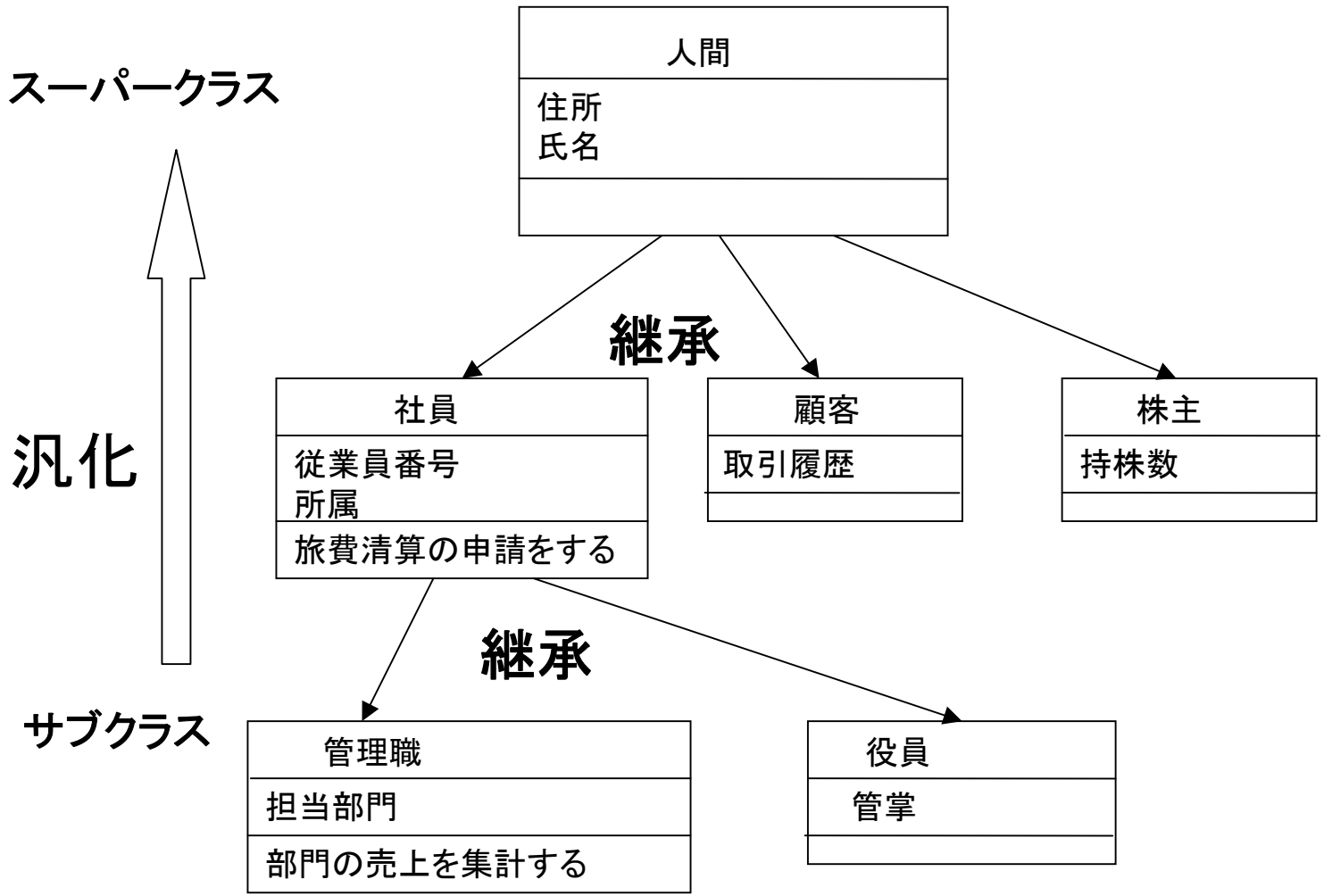
・支店番号
・口座番号
・預金残高
・自動引き落とし項目

・引き落とす　{
　預金残高が引き落とし金額＋手数料よりも多いなら、
　　引き落とし金額を支払い、預金残高から引き落とし金額＋手数料を引く
　少ないなら、総合口座の定期預金額が不足分をカバーできるかチェックする
　　カバーできるなら、不足分をローンにして引き落とし金額を支払う
　　カバーできないなら、エラーメッセージを出す　}
・預金する　{. . . . . }
・残高を照会する　{. . . . . }
・利子をつける　{. . . . . }

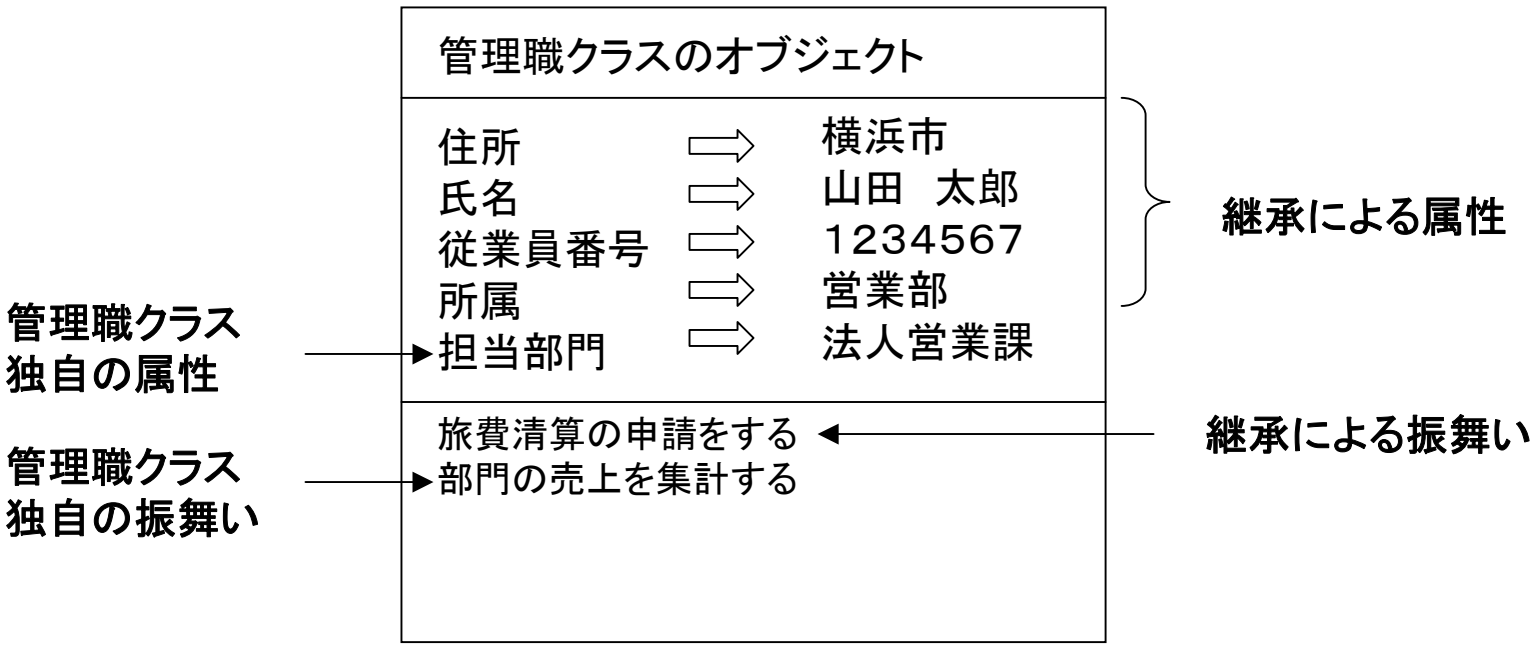
クラス同士のメッセージのやり取り



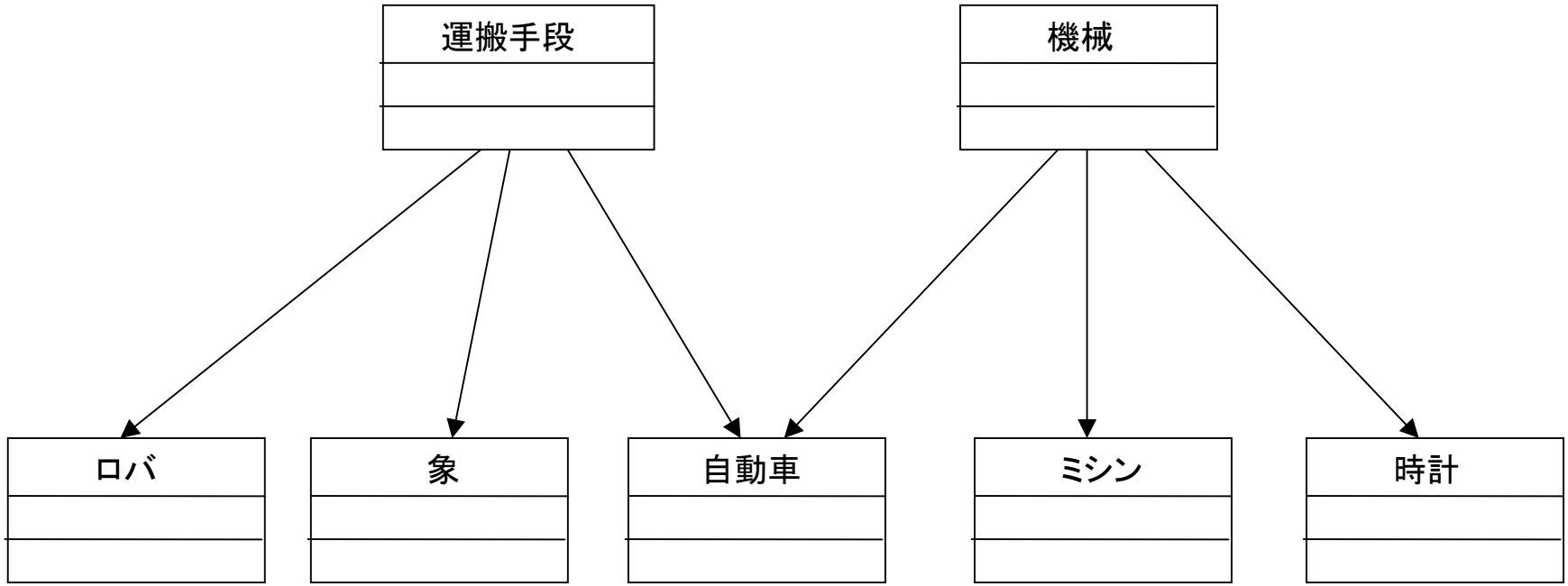
クラスとクラスの親子関係（汎化と継承）



管理職クラスのオブジェクト

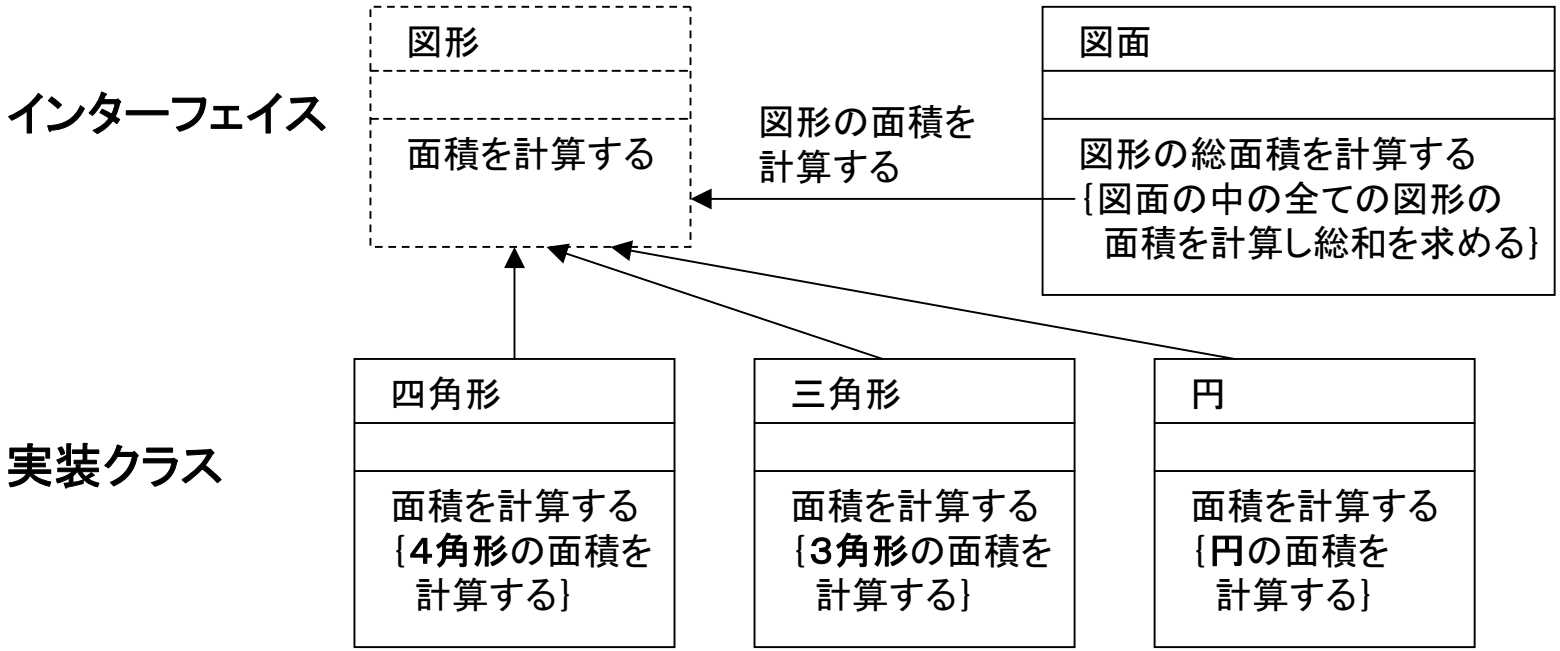


多重継承



ポリモルフィズム

インターフェイスを介して実装にアクセスする機能



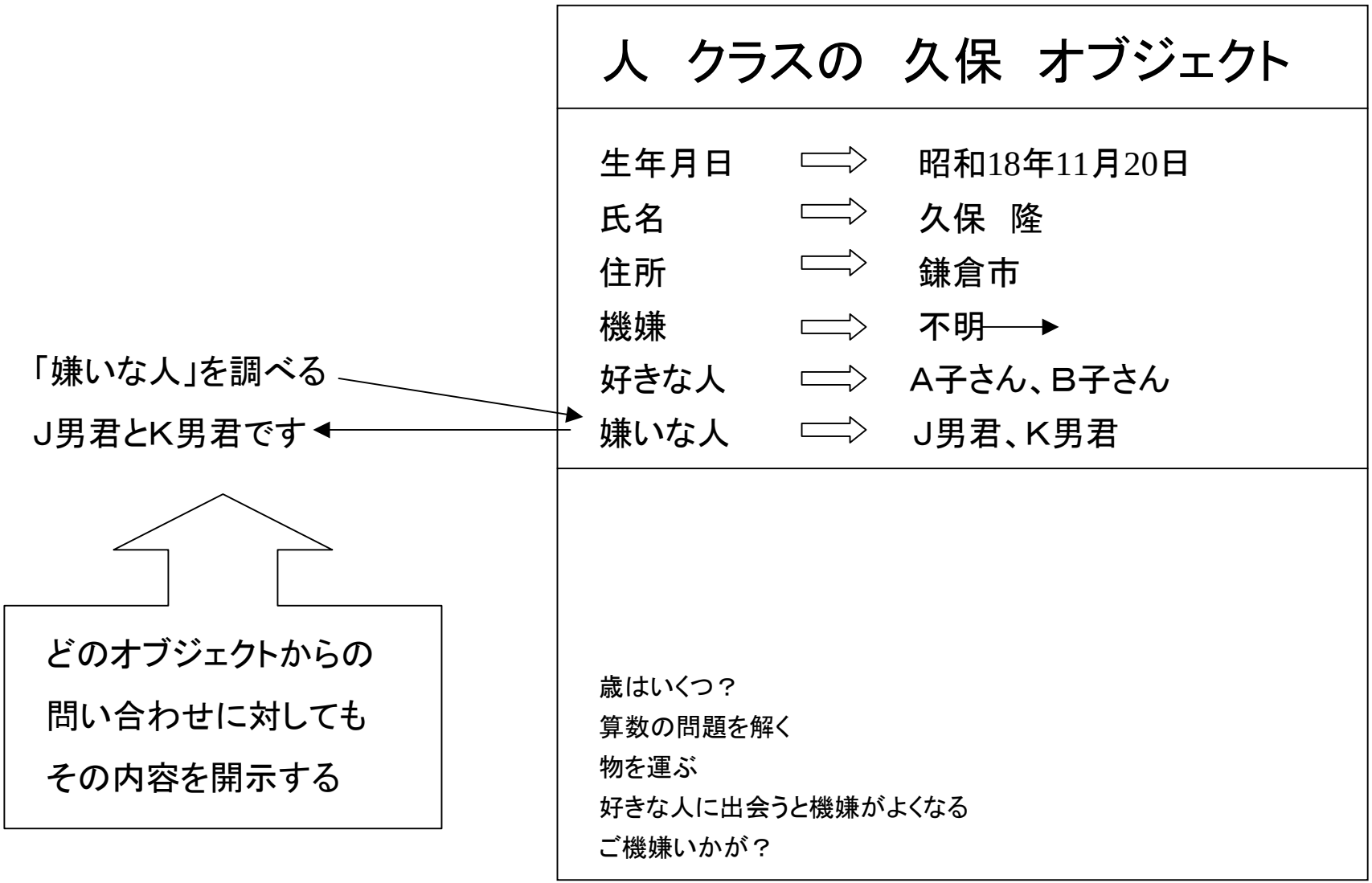
・四角形／三角形／円は図面クラスからは図形インターフェイスを介して参照する。
　従って図面クラスは四角形／三角形／円の区別を意識することはない。
・将来楕円などの図形を追加しても、図面クラスは影響を受けることはない。

隠蔽

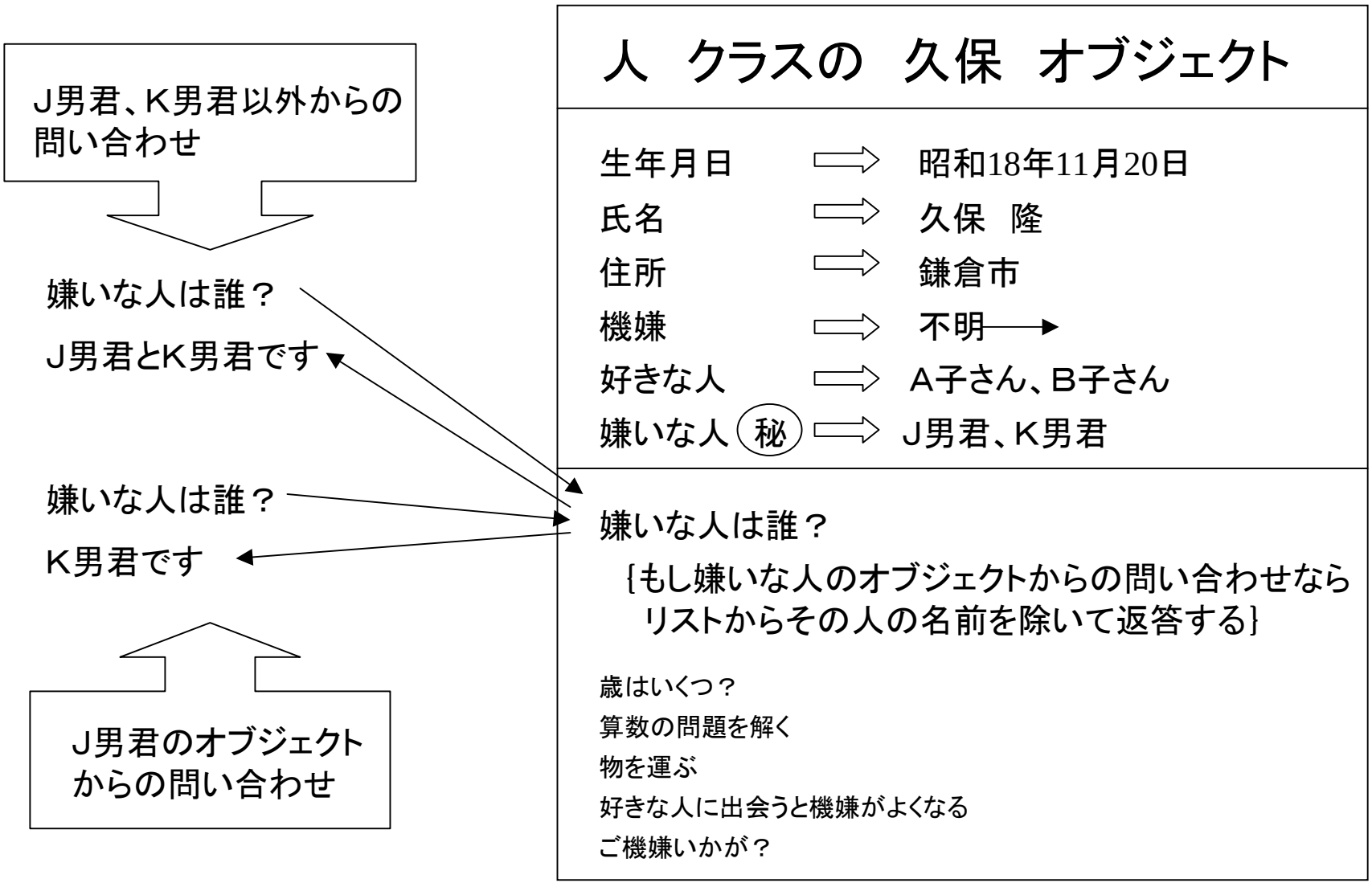
クラスの中の属性や振舞いを、クラスの外部に対して隠す機能

- **何を隠すのか**
  - 「属性」や「振る舞い」を隠す
  - 実装を隠す
- **何から隠すのか**
  - 他の全てのクラスから隠す
  - 子供のクラス以外のほかのクラスから隠す
  - 特定のグループに所属するクラス以外のクラスから隠す
- **なぜ隠すのか**
  - 隠すことによりクラスの情報をガードする
  - 実装を変更しても他のクラスに影響を与えないようにする

隠蔽しない場合


人　クラスの　久保　オブジェクト
生年月日
昭和18年11月20日
氏名
久保　隆
住所
鎌倉市
機嫌
不明
好きな人
A子さん、B子さん
嫌いな人
J男君、K男君
「嫌いな人」を調べる
J男君とK男君です
どのオブジェクトからの
問い合わせに対しても
その内容を開示する
歳はいくつ？
算数の問題を解く
物を運ぶ
好きな人に出会うと機嫌がよくなる
ご機嫌いかが？

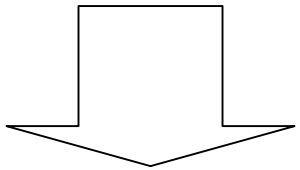
隠蔽する場合


J男君、K男君以外からの問い合わせ
嫌いな人は誰？
J男君とK男君です
嫌いな人は誰？
K男君です
J男君のオブジェクトからの問い合わせ
人　クラスの　久保　オブジェクト
生年月日 ⇨ 昭和18年11月20日
氏名 ⇨ 久保　隆
住所 ⇨ 鎌倉市
機嫌 ⇨ 不明
好きな人 ⇨ A子さん、B子さん
嫌いな人 (秘) ⇨ J男君、K男君
嫌いな人は誰？
{もし嫌いな人のオブジェクトからの問い合わせなら
リストからその人の名前を除いて返答する}
歳はいくつ？
算数の問題を解く
物を運ぶ
好きな人に出会うと機嫌がよくなる
ご機嫌いかが？

オブジェクト指向で分析・設計をすると言うこと

- オブジェクトを見つけ出す
- 似通ったオブジェクトを集め、抽象化をしてクラスを見つけ出す
- クラスとクラスの親子関係を見つけ出す
- クラスとクラスの機能分担を決め、相互の関係を見つけ出す
- クラス同士の相互の関係から、「属性」と「振る舞い」を見つけ出す

  「属性」と「振る舞い」が決まれば、そのクラスがどんなメッセージを受け取れるかが決まる
- 「属性」と「振る舞い」の隠蔽のレベルを決める
- デザインパタンの流用を検討する



オブジェクト指向の言語で実装するのならば、ここからプログラムに落とし込むことができる

結局　オブジェクト指向とは

- 要求分析の技術
- モデリング（設計）の技術
- プログラミングの技術
- 部品化（再利用）の技術

結局　クラス（オブジェクト）とは

- 要求分析・設計の対象
- 実装の対象
- 隠蔽のためのカプセル
- 部品化（再利用）の単位

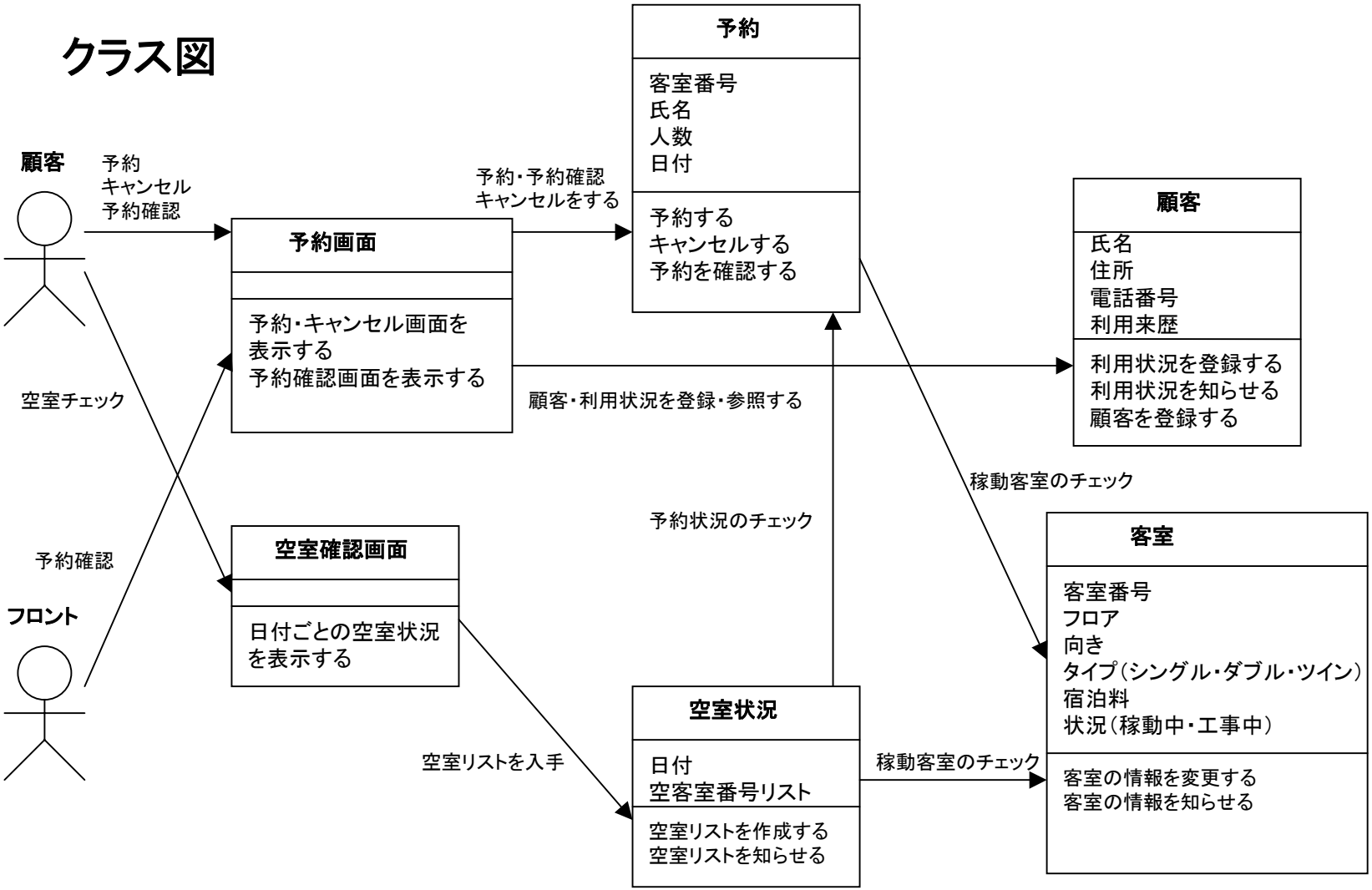
[例] ホテル予約システム

要求仕様

- 顧客は空室確認画面により、予定日ごとの客室の空き状況を　確認できる。
- 顧客は予約画面により、客室の予約ができる。
- 顧客は予約画面により、予約をキャンセルできる。
- 顧客は予約画面により、自分の予約状況を確認できる。
- 顧客のホテル利用来歴が記録されており、予約時に新規顧客とは挨拶が変わる。
- フロントは来客時に、顧客の予約確認ができる。

[例] ホテル予約システム

クラス図



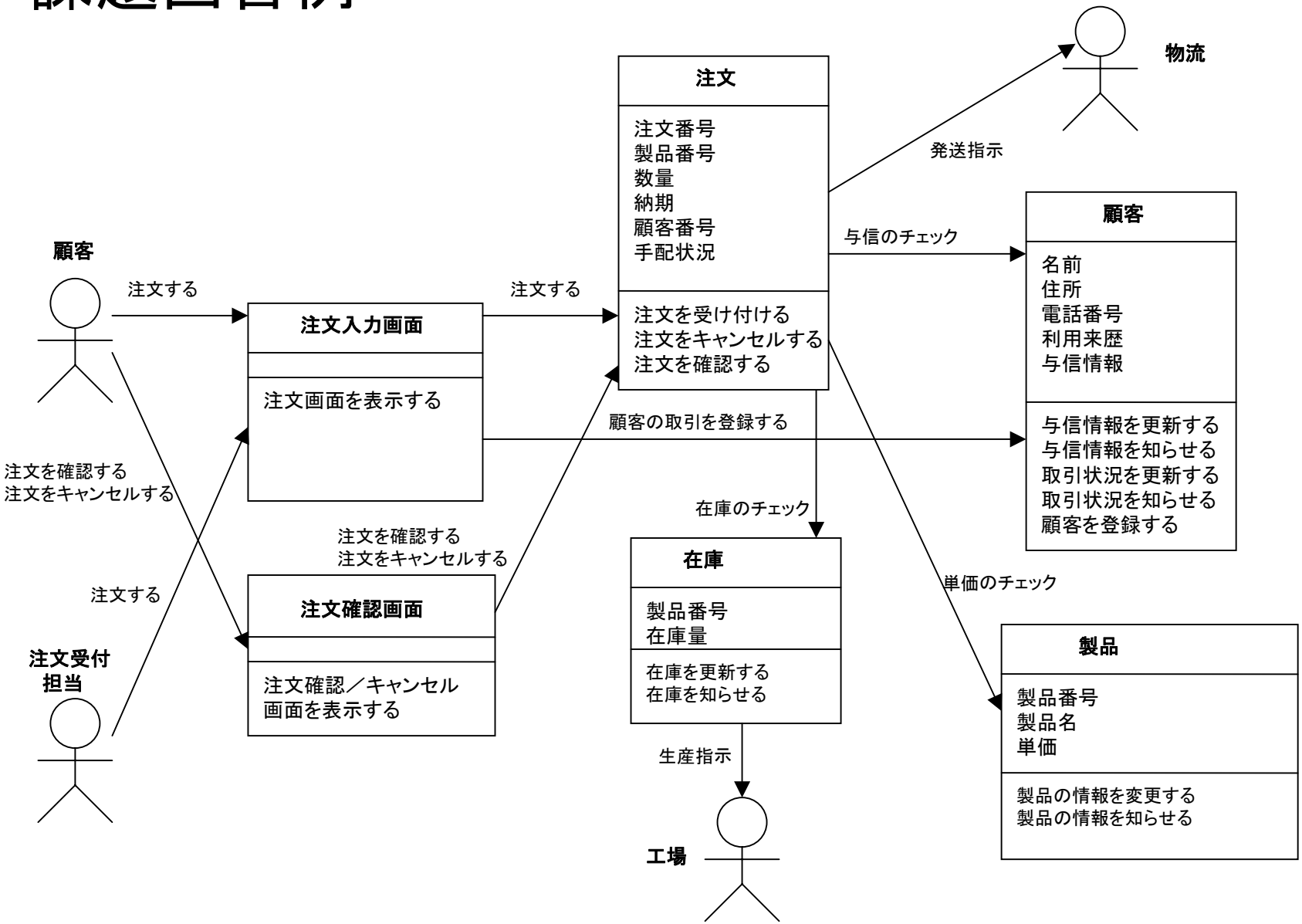
演習課題

以下の要求仕様を満たす受発注システムのクラス図を記述しなさい

要求仕様

- 顧客は注文入力画面により、製品を注文できる。
- 注文受け付け担当者は、電話やFAXで受けた注文を顧客に代わり
  注文入力画面から注文できる。
- 顧客は注文確認画面により、自分の注文製品の状況を確認できる。
- 顧客は注文確認画面により、発送以前の状態の注文をキャンセルできる。
- 顧客からの注文があった場合、与信システムに顧客の信用を照会し、
  OKなら注文を受け付ける。
- 顧客からの注文を受け付けた場合、在庫をチェックし、在庫があれば
  これを引当、発送する。
- 在庫がない場合、工場に生産指示を出し、入庫に伴いこれを引当
  発送する。

課題回答例


物流
注文
注文番号
製品番号
数量
納期
顧客番号
手配状況
注文を受け付ける
注文をキャンセルする
注文を確認する
発送指示
顧客
与信のチェック
名前
住所
電話番号
利用来歴
与信情報
与信情報を更新する
与信情報を知らせる
取引状況を更新する
取引状況を知らせる
顧客を登録する
顧客
注文する
注文する
注文入力画面
注文画面を表示する
顧客の取引を登録する
注文を確認する
注文をキャンセルする
在庫のチェック
注文を確認する
注文をキャンセルする
注文する
注文確認画面
注文確認／キャンセル
画面を表示する
在庫
製品番号
在庫量
在庫を更新する
在庫を知らせる
単価のチェック
製品
製品番号
製品名
単価
製品の情報を変更する
製品の情報を知らせる
注文受付
担当
生産指示
工場

まとめに代えて

- オブジェクト指向の勉強はプログラム言語から入るのが、
  遠回りの様で実は最短。

- SmallTalkから入る場合、お勧めは、**オージス総研**の教材
  http://www.ogis-ri.co.jp/otc/hiroba/technical/Squeak/squeak.html

- C言語の知識（深い知識の必要はない）が有れば、Java言語が
  お勧め。 参考書多数有り。
  短時間で学びたい場合は、Sun社認定の教育コースがお勧め。
  Java プログラミング Ⅰ コース（3日間 ￥128．000）等

- 習うだけでなく、慣れる事が必須