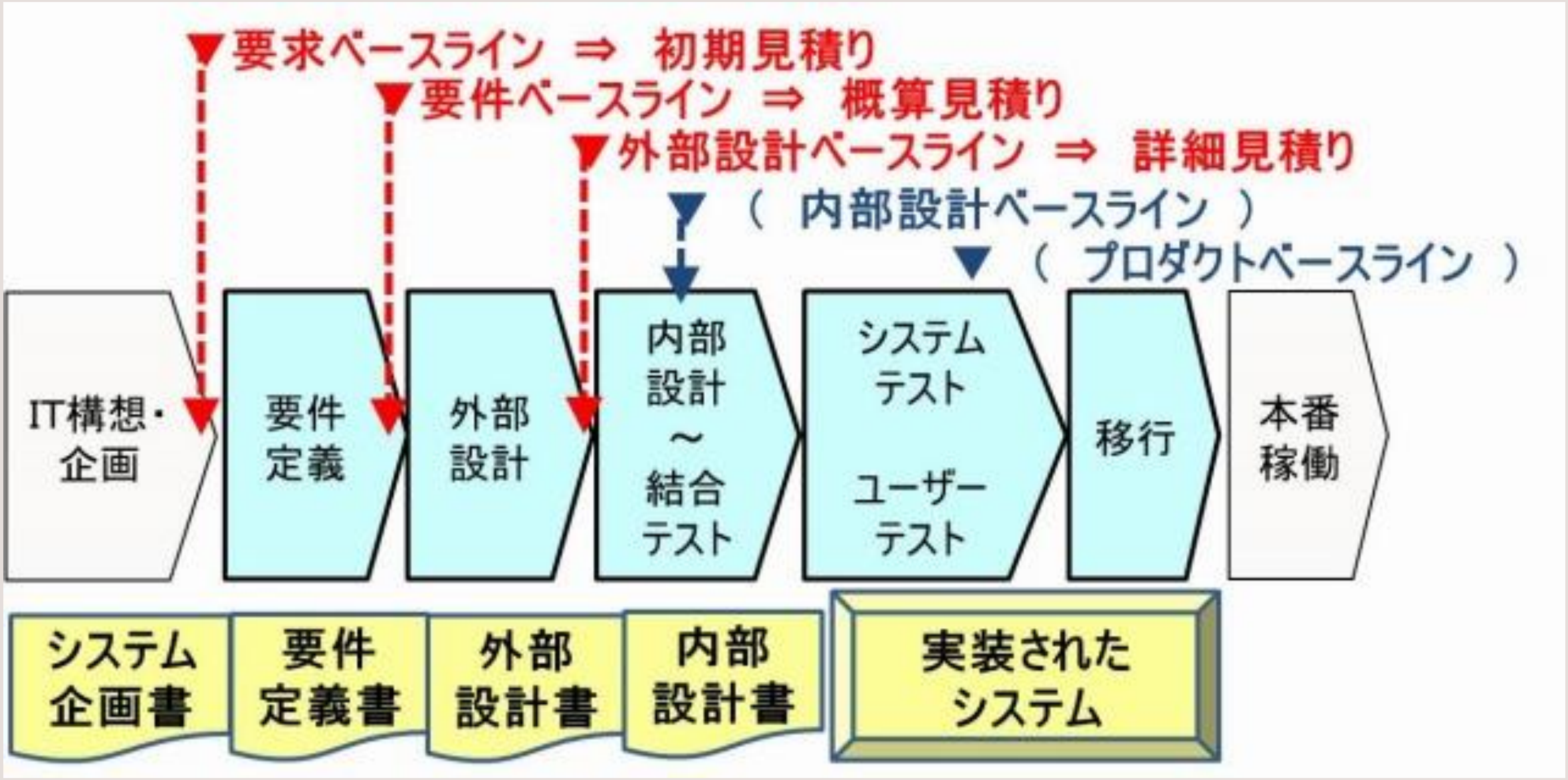
システム設計実践演習
第８回

仕事の流れについて


要求ベースライン ⇒ 初期見積り
要件ベースライン ⇒ 概算見積り
外部設計ベースライン ⇒ 詳細見積り
（ 内部設計ベースライン ）
（ プロダクトベースライン ）
IT構想・企画
要件定義
外部設計
内部設計～結合テスト
システムテスト
ユーザーテスト
移行
本番稼働
システム企画書
要件定義書
外部設計書
内部設計書
実装されたシステム

内部設計
(詳細設計)

詳細設計(内部設計)とは

詳細設計では、システムを「入力→処理→出力」で考えたとき、「処理」に関連する部分を設計します。

つまり、**システム内部の動作を設計**することを考えます。

詳細設計(内部設計)とは

詳細設計で考える項目は、システムによって様々ですが、次の内容がよく挙げられます。

- データベース設計
- ネットワーク設計
- サーバー設計
- **ロジック設計**

データベース設計

データベース設計は、データに関する設計を実施します。

リレーショナルデータベースやNoSQLなどのミドルウェアの選定は、要件定義および基本設計でおおよそ完了しているので

詳細設計では、実際のデータをどのような形で格納するのかを設計します。

データベースの設計書類

分類	設計書名	概要
RDB系	テーブル一覧	テーブル一覧を管理する。サブシステムごとに管理するなど体系立てて管理するとよい。
	ER図	正規化を行い、テーブル同士の関係性を設計し図として整理します。 ※概念レベルは要件定義で実施している
	テーブル レイアウト	各テーブルの項目などを設計。
NoSQL系	データ一覧	データの一覧を管理する
	データ レイアウト	各データの項目などを設計する。 NoSQL型はリレーションできないのでER図では描画不可能

ネットワーク設計

要件定義の非機能要件をもとに、システムで必要となる個々のネットワーク設計を行う。基本的には、**ネットワーク設計は採用した機器やプロダクトがベースにあり、それらの仕様に従ってどのように組み立てるか？どのような設定をするのか？が主となります**。

ネットワークの設計書類

設計書名	概要
ネットワーク 物理構成図	物理的な機器構成の設計図。ネットワーク機器、サーバー、ストレージなどのシステム構成要素を図式化する。
ネットワーク 論理構成図	論理的な構成の設計図。機能的な配置と、その接続を整理した図。アプリケーション担当が参照する資料となる。
機器一覧	設置する機器の一覧表
ネットワーク サービス設計	VPNやDNSといったサービスの具体的な設計を行う。

※ネットワーク機器やプロダクト固有の設計があり、
　詳細設計で行うべき内容はかなり異なります。

サーバー設計

サーバーを構築し、運用するための設計を行います。**ネットワークと同様に、サーバーにおいても機器やプロダクトがベースにあるので、それらを最適に組み合わせていく形になります。**

サーバの設計書類

設計書名	概要
サーバー一覧	サーバーの一覧を整理します。仮想化している場合は、物理サーバーと論理サーバーのどちらも管理する必要がある。
サーバー 仕様設計	サーバーのスペックや台数を明確にします。 また、仮想化の場合はその方式や内容を設計する。
サーバー プロダクト構成	サーバーにインストールする製品を設計する。 また、インストール手順書も合わせて設計する。
サーバー運用 設計	サーバーの継続利用に必要な処理を設計する。また、運用手順書や障害時対応手順なども設計する。

※サーバー機器やプロダクト固有の設計があり、
詳細設計で行うべき内容はかなり異なります。

ロジック設計

ロジック設計は一般的にイメージする「プログラミング」に近い設計になります。採用するソフトウェア設計モデルによる部分もありますが、最後は個々のプログラムのロジックを設計することとなります。

ロジックの設計書類

設計書名	概要
シーケンス図	オブジェクト間の処理のやり取りを時系列で表現する図。
クラス図	オブジェクト指向でいうクラスの関係性を図にしたもの。
プログラム仕様書(処理設計)	ソースコードレベルの処理内容を記載する設計書。

※機能一覧などは、基本設計で精査されているが、必要であれば追記

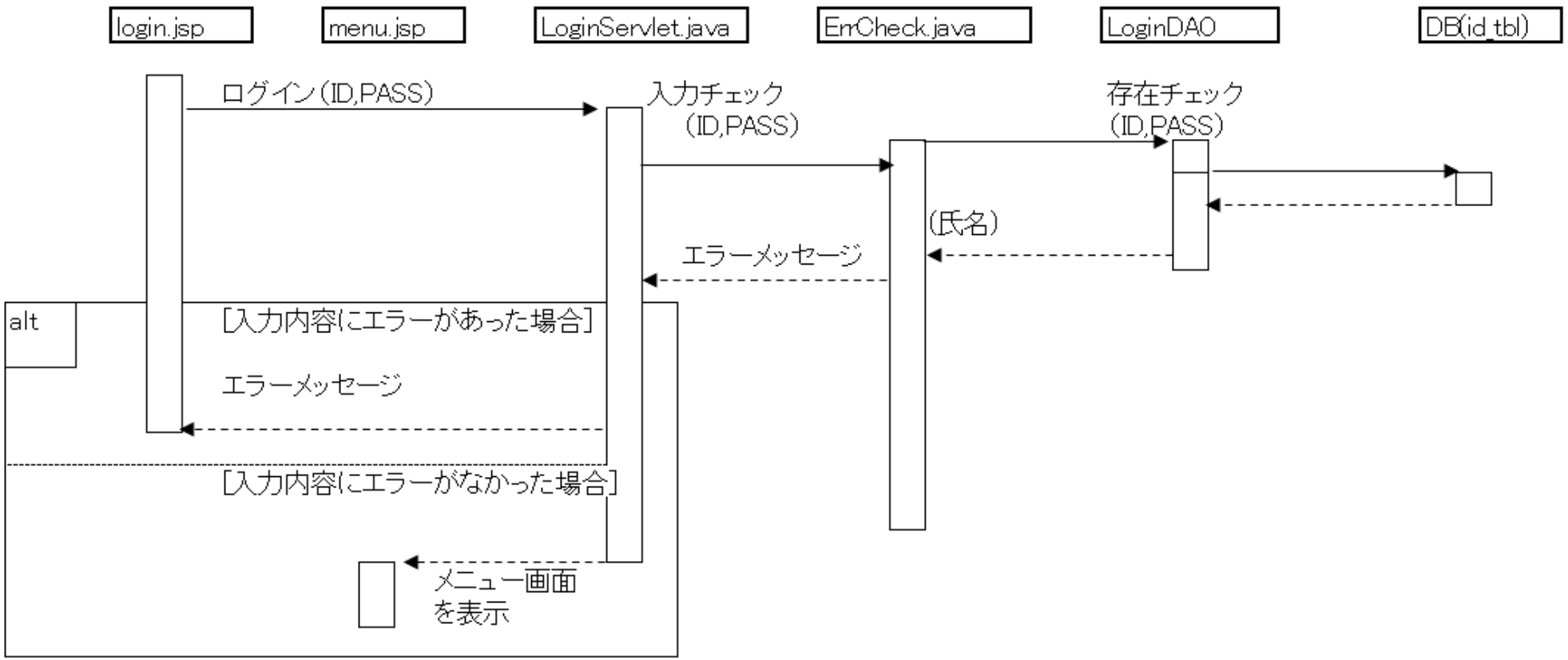
シーケンス図

「**プログラムの処理の流れや概要**」を設計する時に使用されます。**クラスやオブジェクト間のやり取りを時間軸に沿って図にします**。

まず、対象となるシステムや機能を特定し、関連するオブジェクトを書き出します。そしてオブジェクト間で送受信するデータを整理し、時系列順に縦軸に記載していきます。

シーケンス図例

システム名	機能名	開発フェーズ	作成日	作成者	更新日
	ログイン	詳細設計			



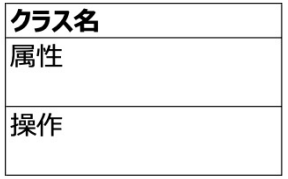
クラス図

クラスの定義や関連性を表す図、主にオブジェクト指向プログラミングで使用されます。

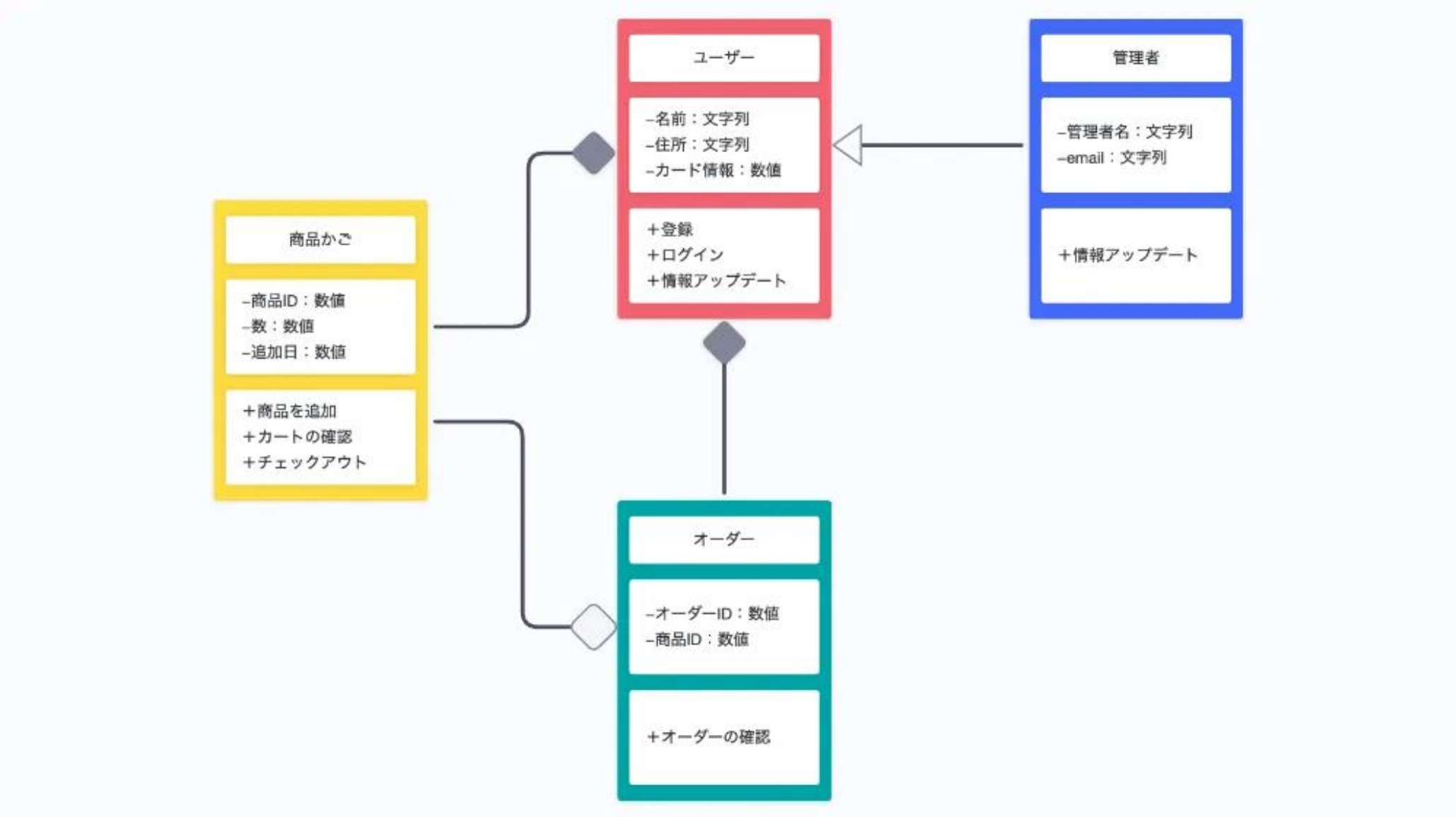
各クラスで定義する値やメソッドをまとめることで、クラスの役割やクラス同士の関連性を把握しやすくなります。

クラス図は「オブジェクト指向」の概念が理解できていないと妥当な設計ができません。その場合は、オブジェクト指向から学習し直しましょう。

クラス図の要素

図	呼び名	意味
	属性	クラスが持つデータ
	操作	クラスが持つ処理
+	パブリック	すべてのクラスからアクセスが可能
−	プライベート	自クラスからのみアクセスが可能
	関連	クラス間で何かしらの関係があるもの
	集約	一つのクラスの中に他のクラスが含まれる関係 （A has-a B：AはBを含む）
	コンポジション	双方のクラスが存在して初めて成り立つ関係 （A part-of B：AはBの一部）
	依存	相手の状態やアクションにて影響を受ける関係
	汎化(継承)	クラスの親子関係。子は親の属性や操作を継承する （A is a B：AはBの1つ、りんご is a 果物。果物が親クラス）
1..*	多重度	1つ以上

クラス図例


ユーザー
-名前：文字列
-住所：文字列
-カード情報：数値
+登録
+ログイン
+情報アップデート
管理者
-管理者名：文字列
-email：文字列
+情報アップデート
商品かご
-商品ID：数値
-数：数値
-追加日：数値
+商品を追加
+カートの確認
+チェックアウト
オーダー
-オーダーID：数値
-商品ID：数値
+オーダーの確認

プログラム仕様書(処理設計)

プログラム仕様書は、プログラムの動作や機能を詳細に記述することで、開発者が具体的な実装に迷いなく取り組むことを目的としています。

まずは、基本設計をベースとして対象となるプログラムについて、必要な機能、入出力などを洗い出します。次に、プログラムの処理の流れを設計し資料に記述します。詳細度は、開発者の熟練度や仕様書を共有する範囲により変わりますが、**この内容をみてプログラミングが可能なレベルに記載する必要があります**。

プログラム仕様書例

基本設計書	画面ID	login.jsp	作成者	前田	修正者	
	画面名	ログイン	作成日	2012/4/1	修正日	

1. 画面レイアウト

～～ 省略 ～～

4. 処理内容

(1) ログインボタン押下

- 入力された社員コードとパスワードで社員テーブルを検索する。
  - 存在しない場合、
    ①「入力された内容が間違っています。」とエラーメッセージを表示する。
    ② 入力されたデータはクリアしない。
  - 存在する場合、
    ①メニュー画面に遷移する。

システムエンジニアとプログラマの関係性

SEとPGはよく設計者と開発者で分けられがちです。

しかし、ここまで学習を進めてきた皆さんは、実際にロジックを組んでいないのに設計を描くことの難しさを理解できたと思います。

「PGだからこれぐらい理解してよ」や「SEが設計したからそのまま製造しよう」のように、相手に頼り切った関係だとプロジェクトの運営に支障をきたすことになります。

お互いにフォローしあえる関係性を築くことが大切です。

システムエンジニアとプログラマの関係性