

# 1. 序章

## 1.1. 情報システムの定義

情報システムとは

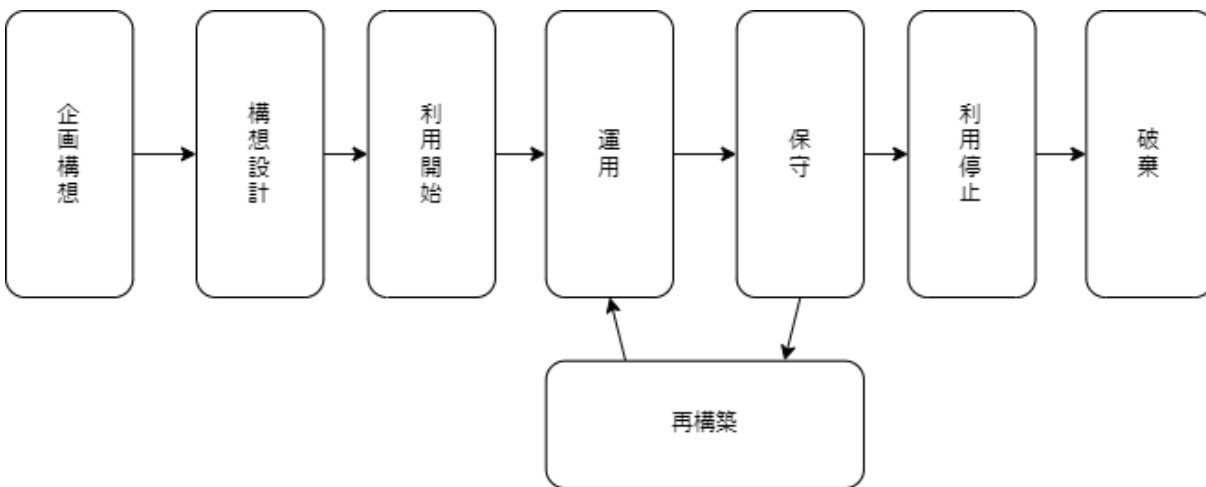
...データ処理システム及び装置であって情報処理を行うもの。

事務機器、通信装置などを含む

今までになかった 情報の処理の伝達などをおこなうプロセスを作ること。

## 1.2. 情報システム開発の目的と目標

## 1.3. 情報システムのライフサイクル



## 1.4. 情報システムの種類とプロジェクト

種類

システム名	説明
生産管理システム	製品の生産計画や生産ラインの管理を行うシステム
原材料調達システム	必要な原材料の調達や在庫管理を行うシステム

製造管理システム	製品の製造工程や品質管理を行うシステム
在庫管理システム	製品や資材の在庫管理を行うシステム
運送管理システム	製品の配送や物流管理を行うシステム
販売管理システム	製品の販売計画や顧客管理を行うシステム
財務会計システム	企業の財務情報や会計処理を管理するシステム

## 利用形態

Government(行政)

Business(企業)

Consumer(個人)

G to C 住民表やパスポートの発行サービスなど

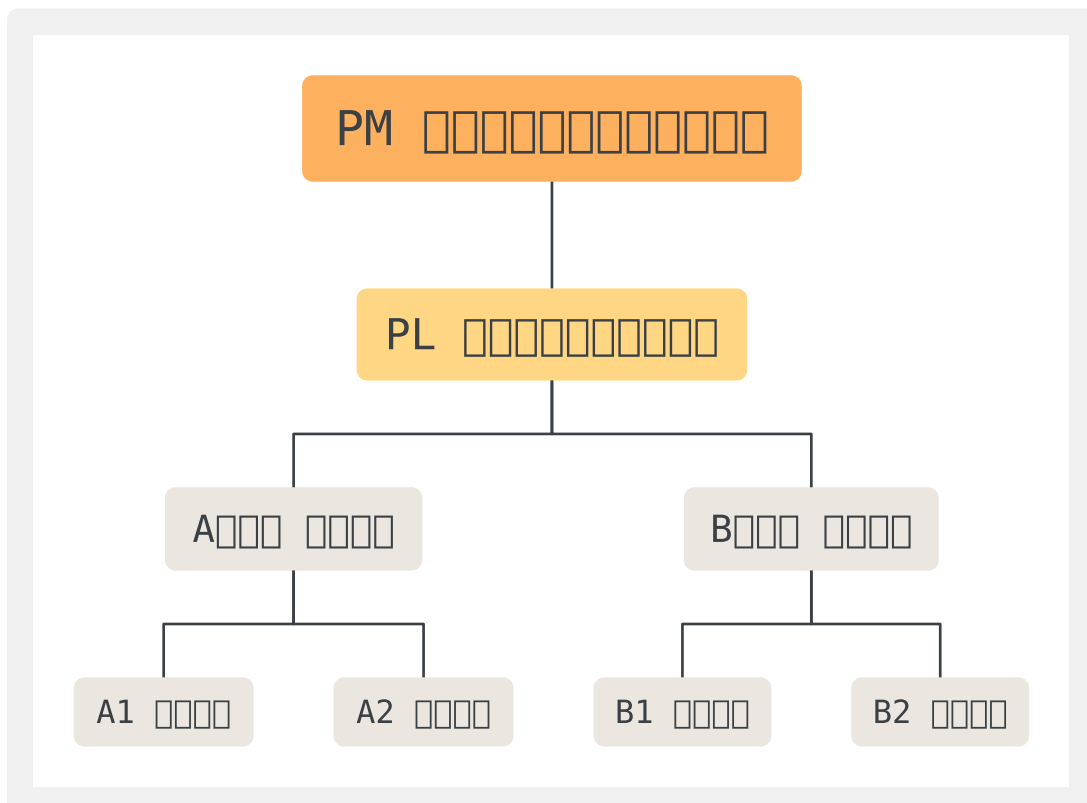
B to G 電子入札システム

B to B 基幹系業務システム

B to C 通販サイト

C to C フリマやオークション

## プロジェクト体制図例



## QCD

...プロジェクトを適切に進めるために適切さを評価する指標

Quality(品質)

Cost(コスト)

Delivery(納期)

## ステークホルダー

...利害関係者

プロジェクトにかかわる企業の経営者や部門の責任者、現場の利用者を指す。

プロジェクト内の「決定権を持つ人」の意味で用いられることもある。

# 2. 情報システムの設計

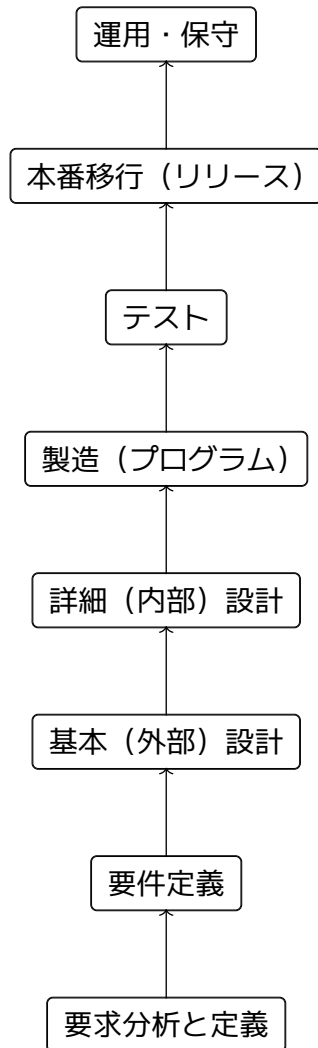
## 2.1. 情報システムの要求分析と定義

---

### 2.1.1. 要求分析と定義の意義

#### 要件定義

...利用者側が実現したいことを明確にしてから「要件定義」として開発者側が作ろうとするものを定義する。



### 2.1.2. 要求定義の活動

#### 意見交換・情報収集方法

- ・ 資料調査
- ・ 観察調査
- ・ ヒアリング

#### 共通理解を進める方法

- ・ 機能要求と非機能要求の確認
- ・ ユースケース図の作成
- ・ 機能改装図の作成

## 要求定義に含まれるもの

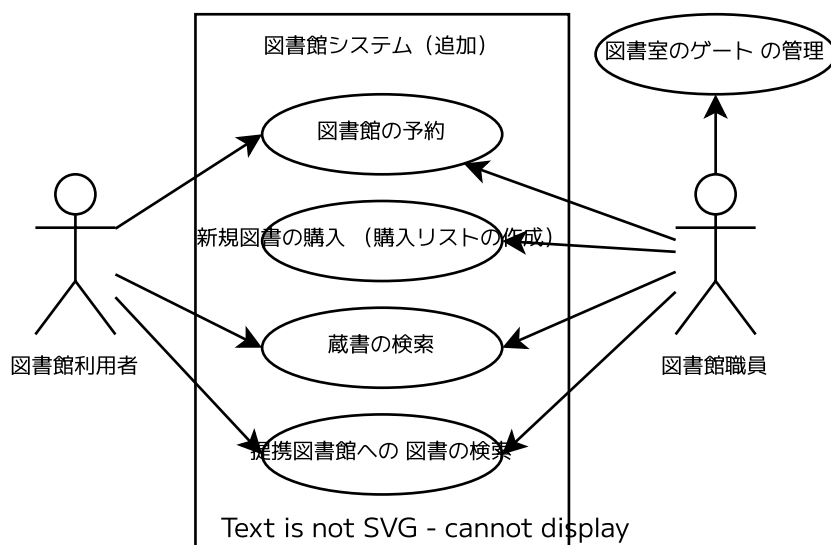
- ・ 背景
- ・ 課題
- ・ 目的
- ・ 概要
- ・ 機能

## ユースケース図

図記号	説明
ユースケース	ユーザーがシステムに対して行う操作、機能
アクター	情報システムを利用する対象、対象の名称
対象	ユースケースを実現する対象

対象内に記述されるユースケースは機能要求を表し、対象外に記述されるユースケースは非機能要求を表す。

P.18 問2

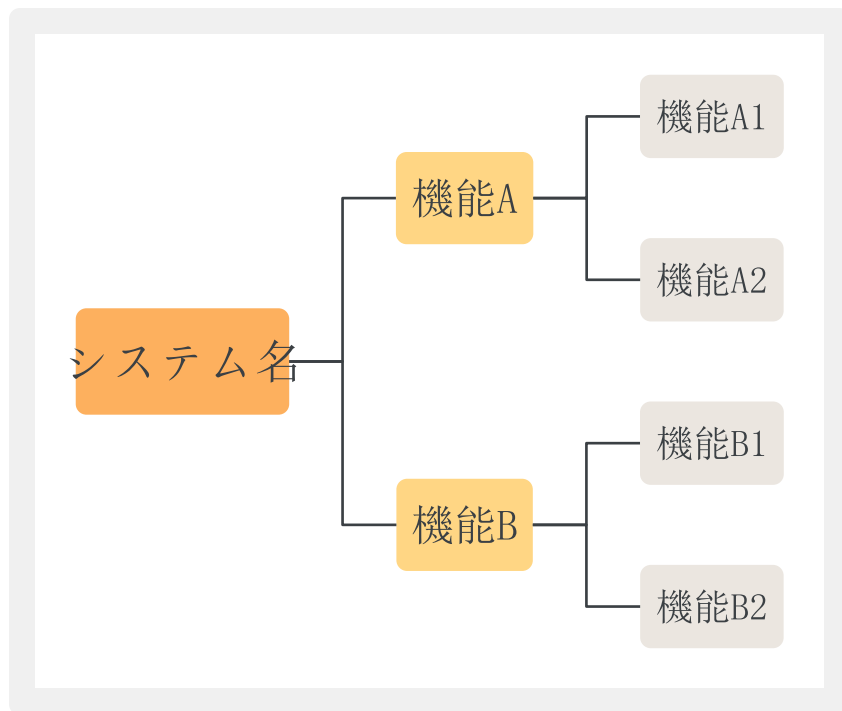


## 機能階層図



### 機能階層図

情報システムが提供する機能にだけ着目し、図示するもの。



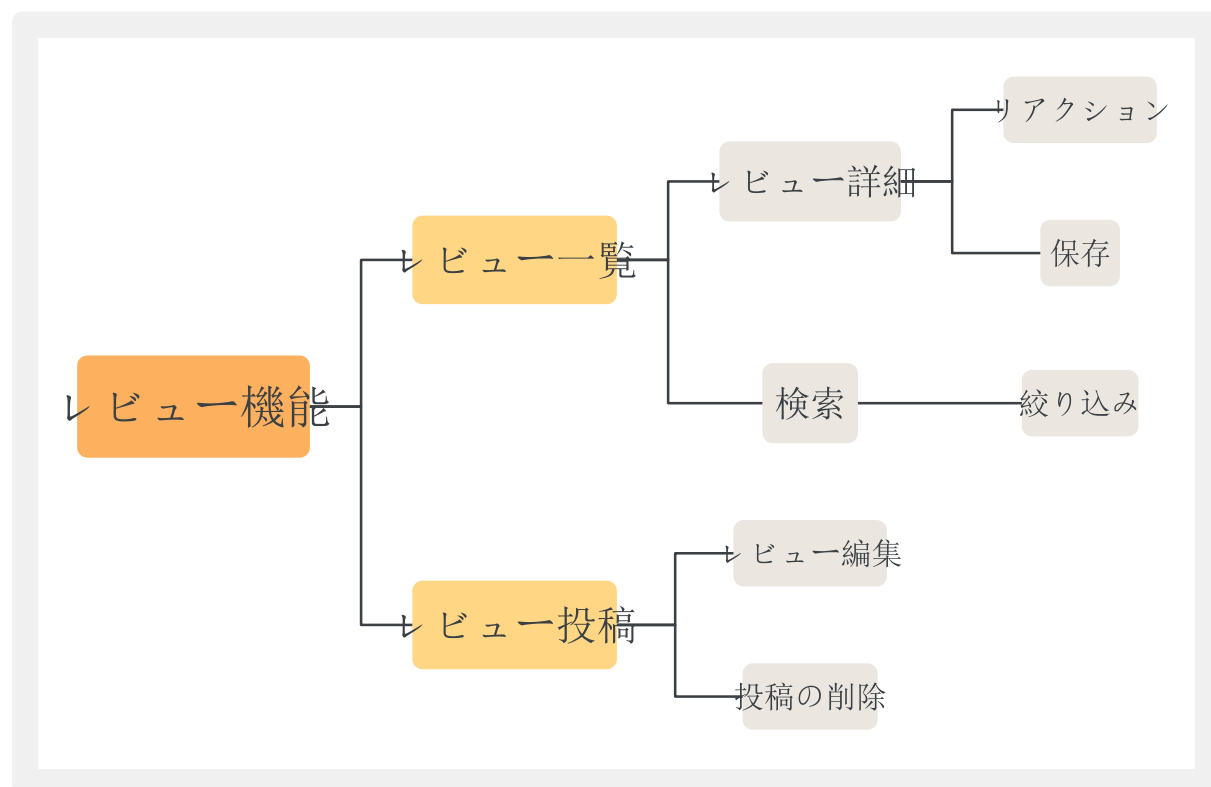
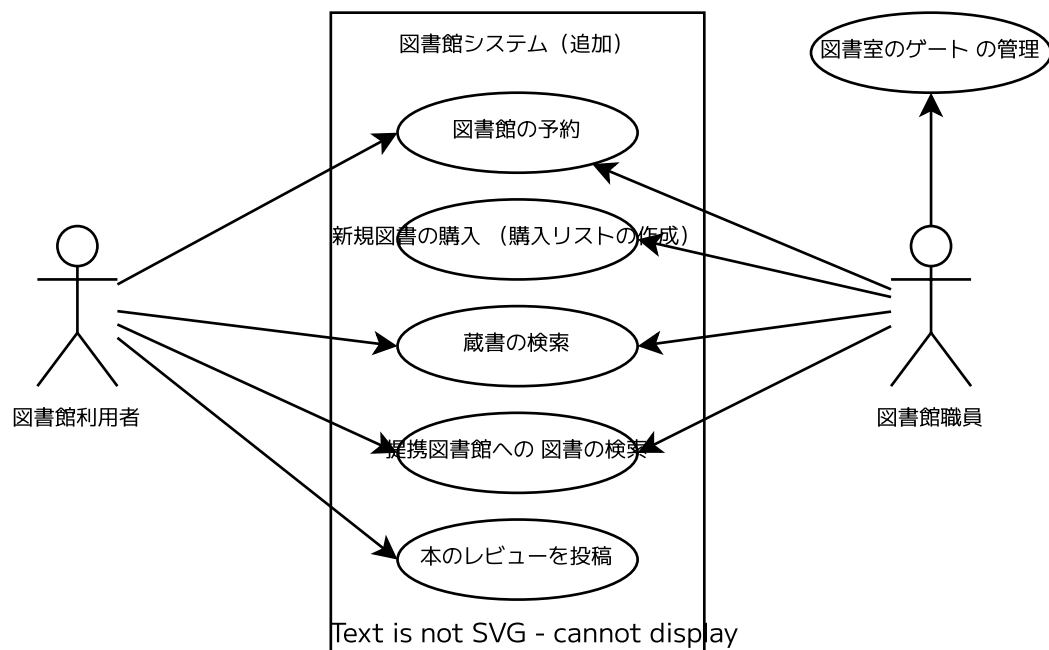
ユースケース図において、明確になった情報システムが備えるべき機能について、さらに細分化した図を作成することができ、設計すべき情報システムのモジュールを明確にすることができる。

### 要求定義

**要求分析** を踏まえて **要求定義** を行う

## 節末問題

2



## 2.2. 情報システムのモデル化

### 2.2.1. システムのモデル化

モデル化

...データの処理の流れを図式化すること

例)

- ・データフロー図 (Data Flow Diagram: DFD)
- ・実体関連図 (Entity Relationship Diagram: ERD)
- ・状態遷移図

### 2.2.2. データフロー図 (DFD)

DFD

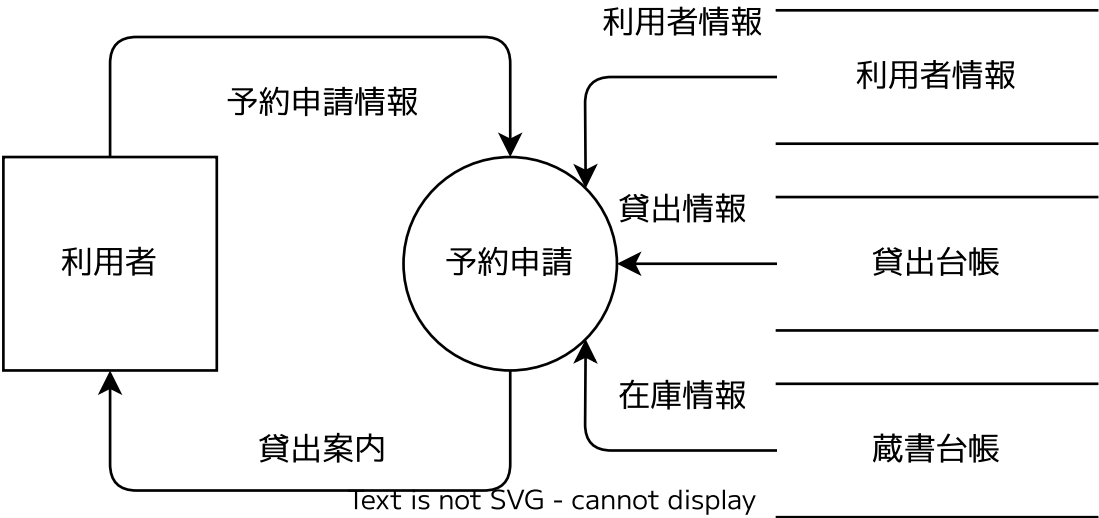
...データの流れを図式化したもの。主に **データフロー**、**データストア**、**外部実体**、**プロセス** の4つの図記号を使って表現される。

名前	図記号	説明
データフロー		データの流れを表す
データストア		データの保管場所を表す
外部実体		システム外部のデータの流れを表す
プロセス		データの加工を表す



例) 図書管理システムにおける「図書の予約」のデータフロー図

図書の予約	
外部実体（発生源、出力先）	利用者
データフロー	予約申請情報、返却情報、予約情報、在庫情報、貸出案内、利用者情報
データストア	利用者台帳、貸出台帳、蔵書台帳
プロセス	予約

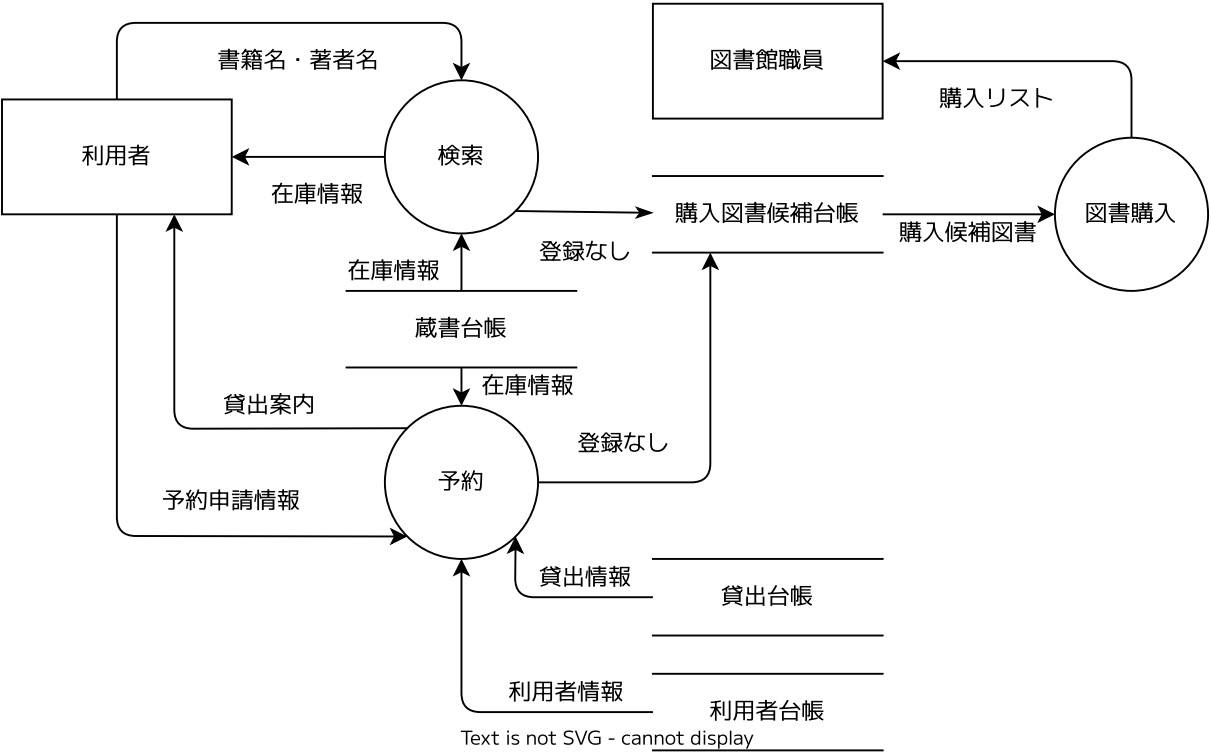


例) 図書管理システムにおける「新規図書の購入」のデータフロー図

新規図書の購入	
外部実体（発生源、出力先）	図書館職員
データフロー	購入リスト、購入候補図書
データストア	購入図書候補台帳
プロセス	図書購入

図書の検索	
外部実体（発生源、出力先）	利用者
データフロー	書籍名・著者名、在庫情報、登録なし（在庫登録なし）
データストア	蔵書台帳、購入図書候補台帳
プロセス	検索

図書の予約	
外部実体（発生源、出力先）	利用者
データフロー	予約申請情報、返却情報、予約情報、在庫情報、貸出情報、登録なし（在庫登録なし）、利用者情報
データストア	利用者台帳、貸出台帳、蔵書台帳
プロセス	予約



## 実体関連図(ERD)

...データ構造を実体（エンティティ）とその間の関係（リレーションシップ）で表現したもの。

名前	説明
実体	データのまとまりを表す。記号の中に実体名を記述する。
関連	実体同士の関連を示す。記号の中に関連名を記述する。
属性	実体を持つ属性情報を表す。記号の中に実体名と属性情報を記述する
カーディナリティ	関連情報の詳細を表す。

名前	説明
1 : 1	ひとつの A に対して B は一つ存在し、逆に一つの B に対して A が存在する関係を表す
1 : n	ひとつの A に対して複数の B が存在し、逆に一つの B に対して A が存在する関係を表す
n : m	複数の A に対して複数の B が存在する関係を表す

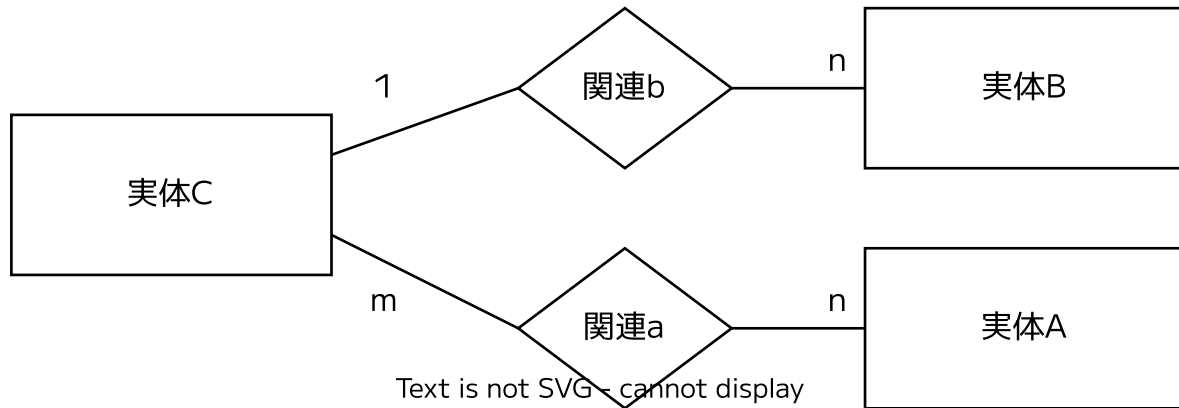
## 現行の図書管理システムにおける実体関連図を作成する

1. 実体と関連を洗い出す
2. 実体と関連のカーディナリティを洗い出す
3. 実体の属性を洗い出す

実体	関連
利用者と図書	貸出
	返却
図書館職員と図書	登録

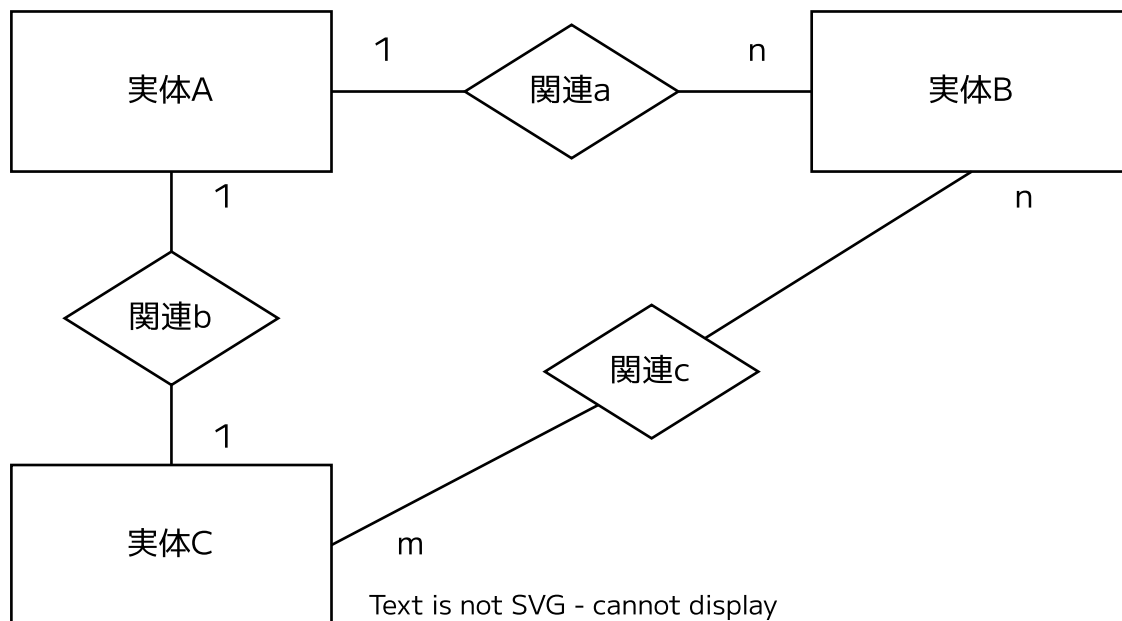
p31 問5

- ・ ひとつの実体 C に対して実体 B は複数存在し、逆に複数の実体 B に対しては実体 C が一つしか存在しない。実体 C と実体 B の関連は「関連 b」とする。
- ・ 複数の実体 A に対して実体 C は複数存在し、逆に複数の実体 C に対しても実体 A は複数存在する。実体 A と実体 C の関連は「関連 a」とする。



p36 ③

次の実体関連図について、それぞれの実体と関連がどのようなになっているか説明しなさい。



- ・ ひとつの実体 A に対して実体 B は複数存在し、逆に複数の実体 B に対して実体 A は一つしか存在しない。実体 A と実体 B の関連は「関連 a」とする
- ・ ひとつの実体 B に対して実体 C は複数存在し、逆に複数の実体 C に対しても実体 B は複数存在する。実体 B と実体 C の関連は「関連 b」とする
- ・ ひとつの実体 C に対して実体 A はひとつだけ存在し、逆にひとつの実体 A に対しても実体 C はひとつだけ存在する。実体 C と実体 A の関連は「関連 c」とする

#### 状態遷移図

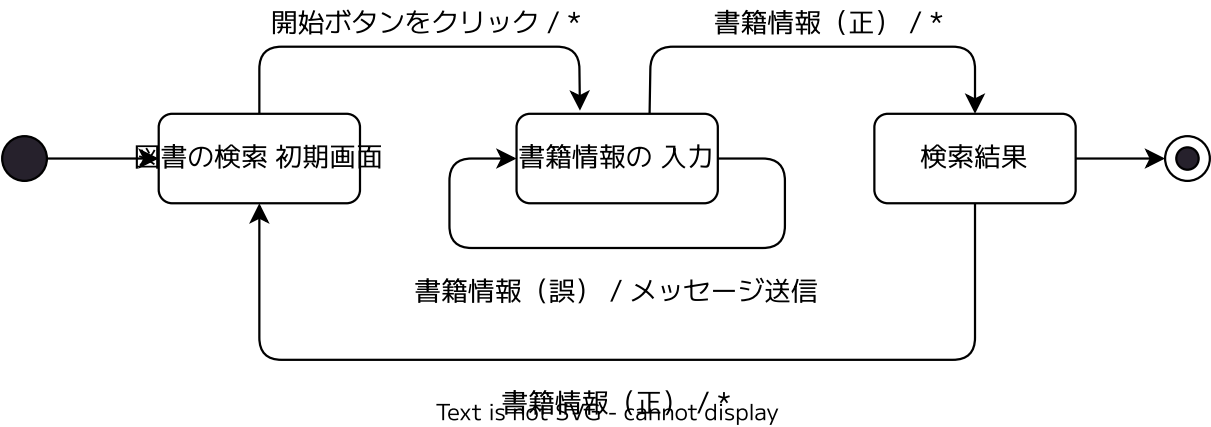
...システムの状態遷移を表す図。状態の移り変わるきっかけ（イベント）とその時に実行する動作（アクティビティ）を合わせて表現する。主に **状態**、**遷移**、**開始状態**、**終了状態** の4つの図記号を使って表現される。

名前	図記号	意味
状態	<div>状態</div>	開始状態、終了状態、状態
遷移	→	状態間の移り変わりを表す
開始状態	●	状態遷移図の開始を表す
終了状態	⊙	状態遷移図の終了を表す

状態遷移表を作成することによって、情報システムの設計を進めるときに、テストの段階などで修正のリスクを回避することができる。

p36 問 6

「図書の検索」処理の端末における入出力時のメニュー画面の状態遷移図を「状態」「遷移（イベント / アクティビティ）」を考えて作成しなさい。



## 3. 情報システムの分割

### 3.1. モジュール分割

#### モジュール

...複雑な機能をできるだけ単純なモジュール（要素）に分割して効率よく設計を進めていくこと。



#### 利点

- ・ 作業をモジュールごとに分担して進めることができる
- ・ 作業中の不具合をモジュール単位で修復作業ができる
- ・ モジュール自体をほかの情報システムで再利用することができる

モジュール分割の技法	
技法	説明
STS 分割	データの流れに注目し、実行すべき機能を入力処理機能（源泉： Source ）、変換機能（変換： Transformation ）、出力機能（吸収： Sink ）の三つに分割する技法
トランザクション分割	トランザクション（データの種類とその処理単位）に応じて分割する技法
共通機能分割	共通する機能をまとめてモジュール化する技法

### 3.2. STS 分割

#### 分割の手順

1. DFD を参考に、データの流れに注目する
2. **最大抽象入力点** と、 **最大抽象出力点** を特定する  
ここでの最大抽象入力点とは、その点以降は入力データが発生しない最大中小最大抽象出力点とは、その点以降は出力データが発生しない最大の場所のことを言う。
3. STS（入力処理機能、変換機能、出力機能）に分割する
4. モジュールに変換する

