

Webシステムプログラミングb 第12講目

到達目標

■ 到達目標

- Webシステムを設計し、開発できること
 - 最終成果物：学生生活で利用できるWebシステム

■ 授業計画

- 第09講 (12/03) : Webシステムの基本設計 (1)
- 第10講 (12/10) : Webシステムの基本設計 (2)
- 第11講 (12/17) : WebシステムのDB設計
- 第12講 (01/07) : WebシステムのUI設計
- 第13講 (01/21) : Webシステムの処理
- 第14講 (01/23) : 補講日、サンプル課題設計書
- 第15講 (01/28) : 平常試験（成果物確認）、まとめ

第11講：WebシステムのDB設計

■ 講義内容

■ DB設計

- 必要なデータの洗い出し、正規化
- ER図、テーブル定義表

■ レポート課題

■ 学生生活で利用できるWebシステムのDB設計

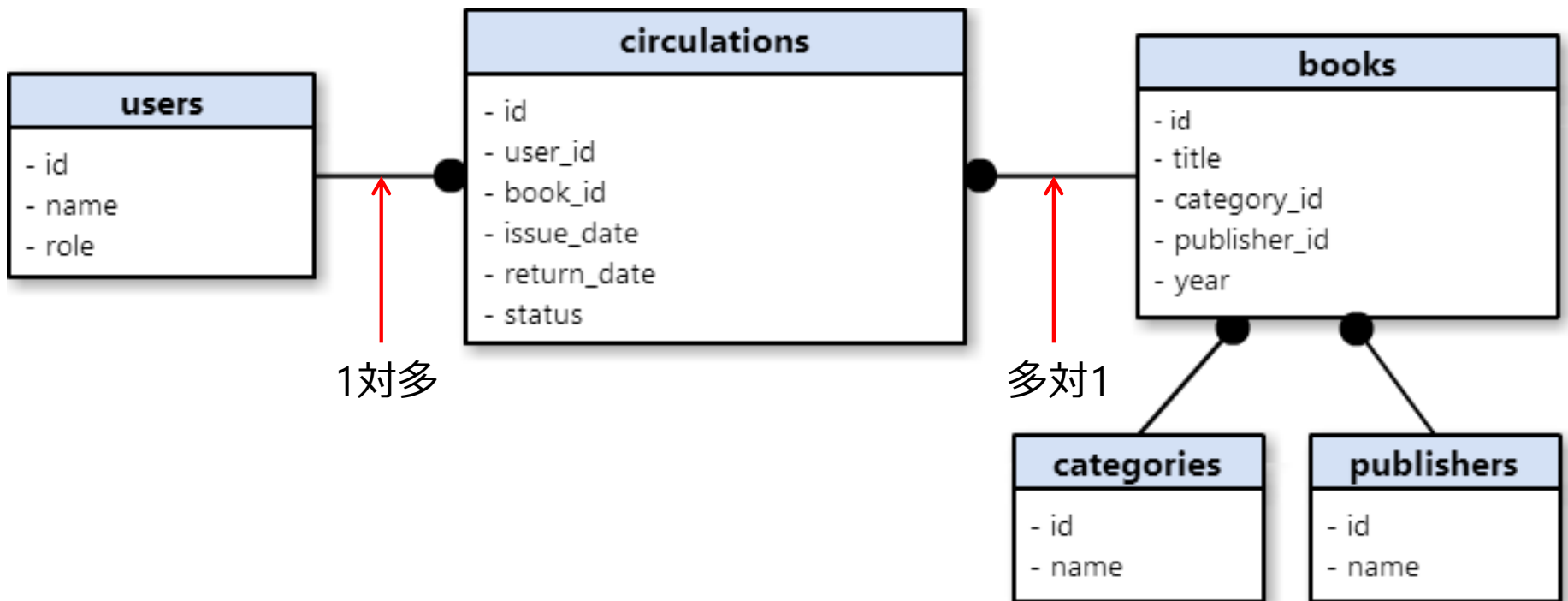
- ER図、テーブル定義表

■ 要件

- DBには学生テーブルを含む3つ以上のテーブルがあること
- 各テーブルは第三正規形になっていること

レポート作成例：ER図

- サンプル課題：以下の5テーブルで構成される
 - users：利用者テーブル, books：蔵書テーブル
 - categories：カテゴリテーブル, publishers：出版社テーブル
 - circulations：貸出テーブル



レポート作成例：テーブル定義表

表. 蔵書テーブル：books

フィールド名	型	Not null	Key	Auto increment	Default	意味
id	INT(11)	YES	主キー	YES	NULL	蔵書ID
title	VARCHAR(40)				NULL	書籍名
category_id	INT(11)	YES	外部キー		NULL	カテゴリID
publisher_id	INT(11)	YES	外部キー		NULL	出版社ID
year	INT(4)				NULL	発行年

前回レポートのポイント

■ ER図

- 第三正規形まで正規化してあること
 - テーブルが1つしかないのはNG
 - 例：ゲームレビューならジャンルと会社は別テーブルに分ける
 - 繰り返し項目を排除する
- 1対多の関係が分かるように図示する

■ テーブル定義表

- 各テーブルに必ず主キーがあること
- 外部キーは他のテーブルの主キーであること

DBの正規化補足説明 (1)

■ 正規化されていないテーブル

表. 蔵書テーブル

ID	タイトル	カテゴリ	出版社	発行年
1	Webエンジニアの教科書	理工系	シーアンドアール研究所	2015
2	たのしいRuby	理工系	ソフトバンククリエイティブ	2010
3	嫌われる勇気	心理学	ダイヤモンド社	2013
4	PHP+MySQLマスターブック	理工系	マイナビ	2014
5	量子力学で生命の謎を解く	物理学	ソフトバンククリエイティブ	2015

繰り返しが発生するフィールドは正規化

DBの正規化補足説明 (2)

■ 正規化したテーブル

表. 蔵書テーブル

ID	タイトル	カテゴリID	出版社ID	発行年
1	Webエンジニアの教科書	1	1	2015
2	たのしいRuby	1	2	2010
3	嫌われる勇氣	2	3	2013
4	PHP+MySQLマスターブック	1	4	2014
5	量子力学で生命の謎を解く	3	2	2015

表. 出版社テーブル

ID	名前
1	シーアンドアール研究所
2	ソフトバンククリエイティブ
3	ダイヤモンド社
4	マイナビ

表. カテゴリテーブル

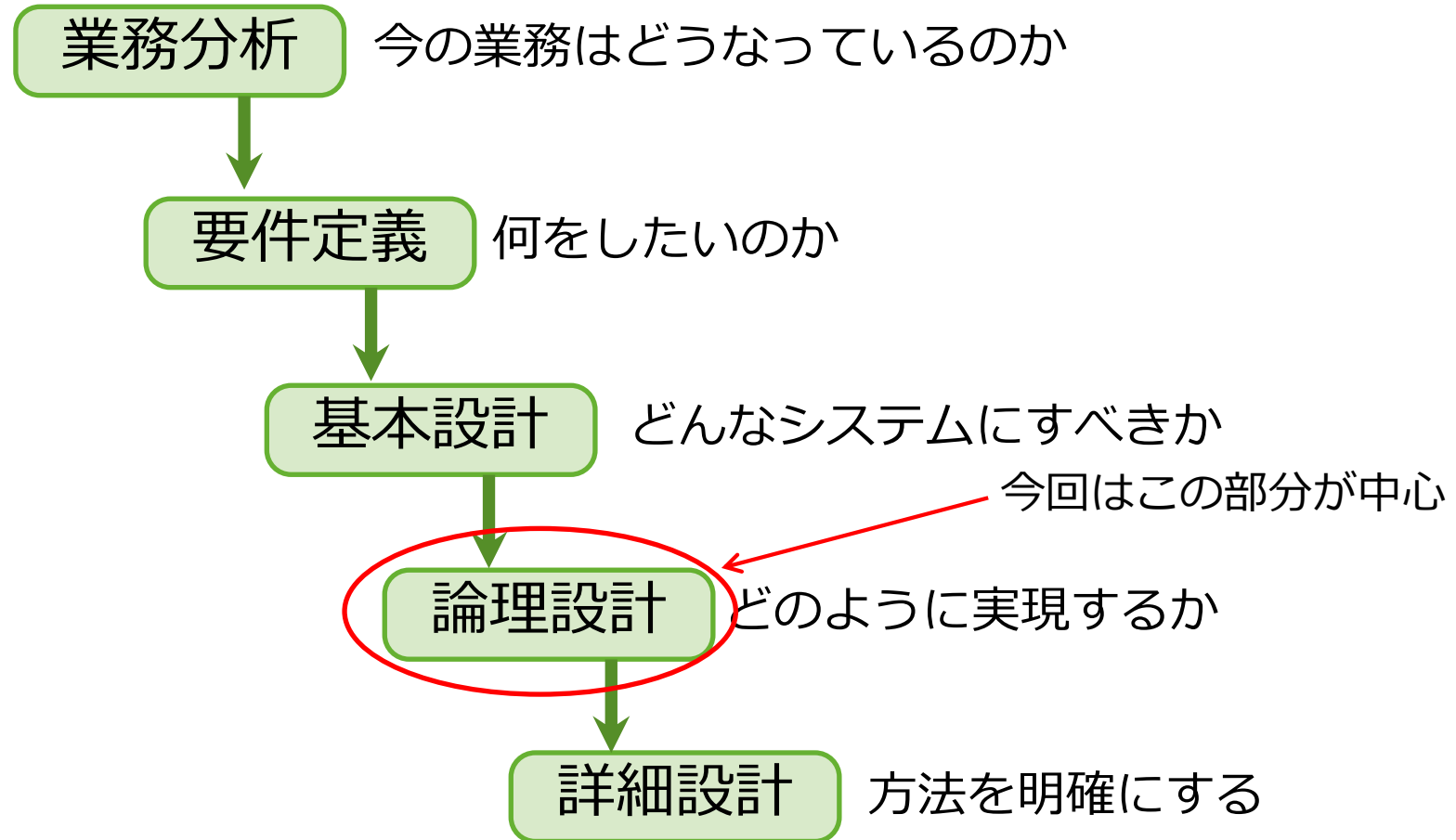
ID	名前
1	理工系
2	心理学
3	物理学

DBの正規化補足説明 (3)

- 正規化が必要な理由
 - データの矛盾を防ぐ
 - フィールドの内容に変更があった場合の修正漏れを防ぐ
 - 例：出版社名に変更があった場合
 - DB処理の効率化
 - フィールドの内容に変更があった場合の修正は1箇所のみでOK
 - 同じ項目（例：同じ出版社）のレコードを全検索する場合の高速化
 - カテゴリの再編時の手間を削減する
 - 例：理工系と物理学は理系に統合する場合
⇒カテゴリテーブルの修正のみでOK
- 必ず第三正規形にすること

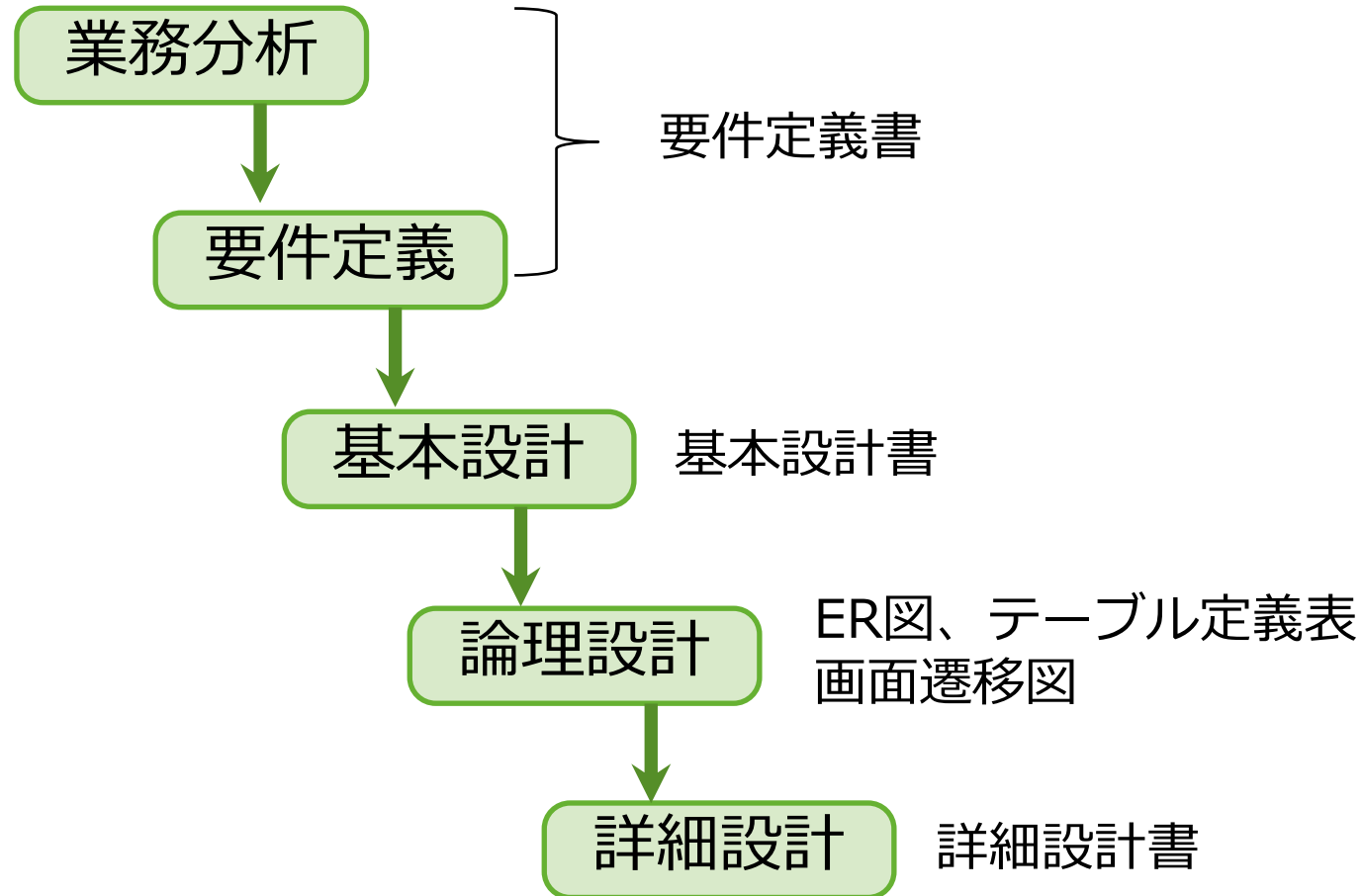
システム設計の流れ

■ DB設計はシステム設計の一部



各工程での成果物

■ 成果物



第12講：WebシステムのUI設計

■ 講義内容

- UI設計：画面遷移図
- DBの実装：DB設計に対応するSQL文

■ レポート課題

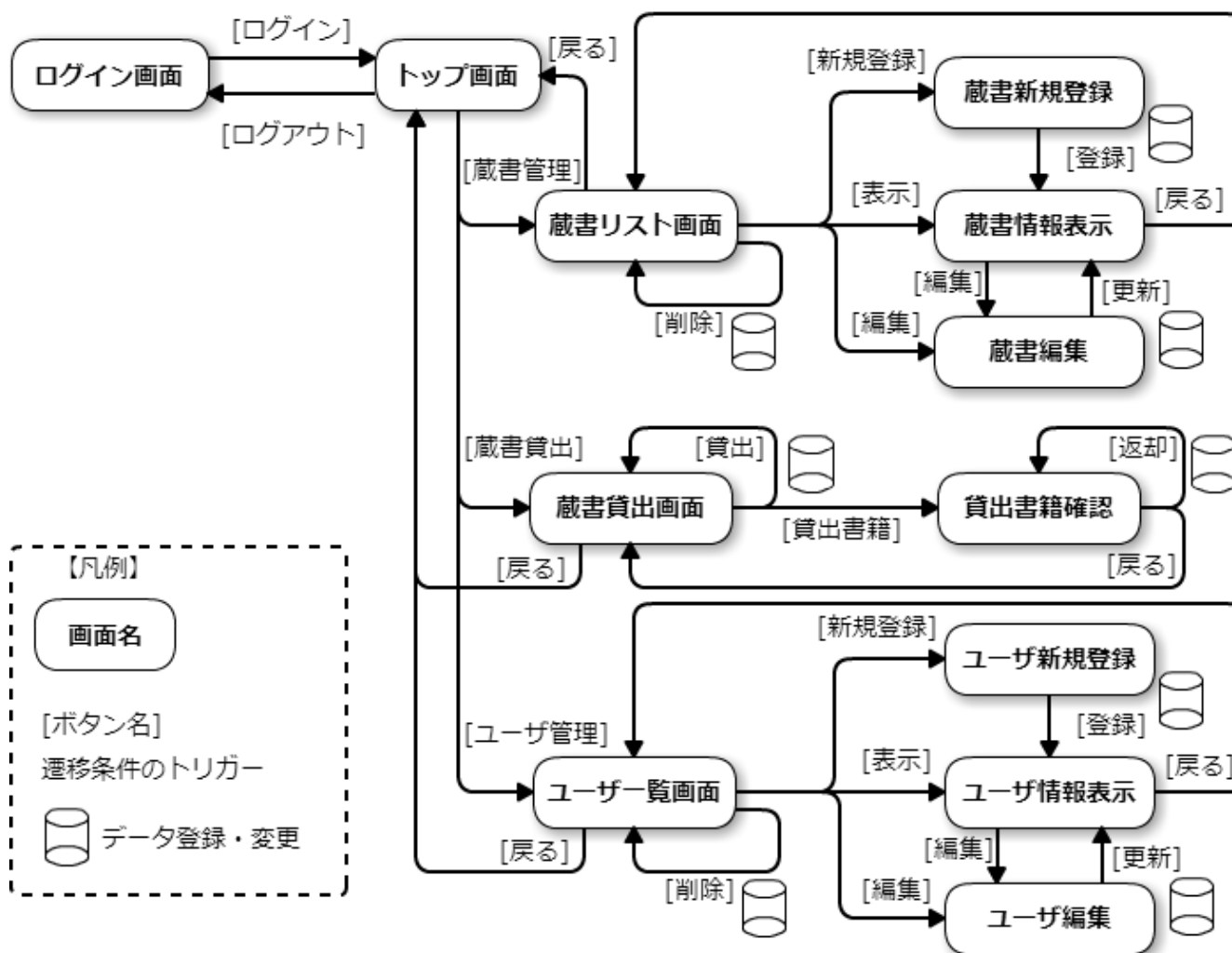
- 学生生活で利用できるWebシステムのUI設計
 - 画面遷移図
 - 第12講目で提出した画面設計図に変更があれば再度提出

画面遷移図

- 業務フローに従い画面遷移とその条件を図示する
- 構成要素
 - 画面ID／画面名
 - 画面を一意に識別する識別子（画面一覧やレイアウトの照合用）
 - 遷移矢線
 - 画面の順序関係を記述。交錯しないように配置
 - ボタン名（イベント）
 - 画面遷移を起こすトリガーを記述
 - ボタン名を記述しておけば、押下時に画面遷移することが分かる
 - アクション（データの発生やメール配信など）
 - 画面遷移に伴って起動される動作を記述
 - データ登録やメール配信のアクションを記述

レポート作成例：画面遷移図

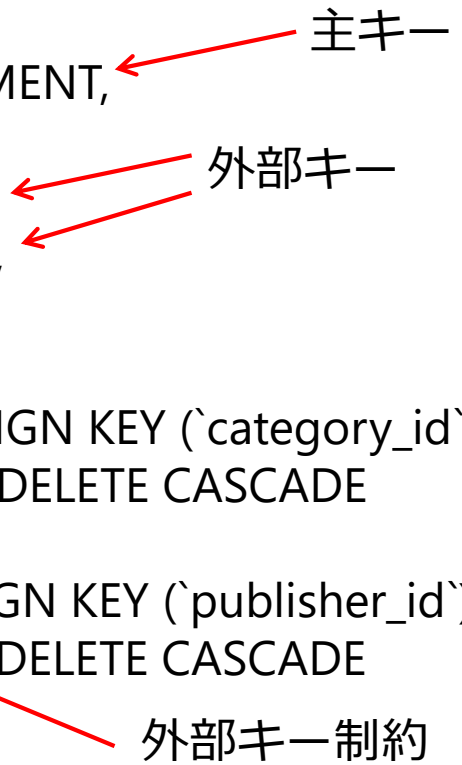
ゼミ室蔵書管理システムの画面遷移図



DB設計とSQL文の対応 (1)

■ 蔵書テーブル (books) のSQL文

```
CREATE TABLE `books` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `title` varchar(40) DEFAULT NULL,  
  `category_id` int(11) DEFAULT NULL,  
  `publisher_id` int(11) DEFAULT NULL,  
  `year` int(4) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `category_id_fk` FOREIGN KEY (`category_id`)  
    REFERENCES `categories` (`id`) ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `publiser_id_fk` FOREIGN KEY (`publisher_id`)  
    REFERENCES `publishers` (`id`) ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```



主キー

外部キー

外部キー制約

※ CASCADE : 親テーブルの更新
に合わせて更新

DB設計とSQL文の対応 (2)

■ カテゴリテーブル (books) のSQL文

```
CREATE TABLE `categories` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(40) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
);
```

■ 出版社テーブル (publishers) のSQL文

```
CREATE TABLE `publishers` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(40) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
);
```


開発について

- 開発言語の指定：とくになし
 - PHP, Java, Ruby, JavaScript, PythonなんでもOK！
 - PHP, Javaだと教員からのアドバイスは得やすい
 - DBはSQL系
- 想定される構成例
 - 画面のフォーム：HTML, JSP
 - DB登録処理：PHP, Servlet
 - DB：MySQL

次回予告

■ 講義内容

- 内部処理の説明
- 授業時間の大半は開発 & 質問の時間

■ レポート課題

- 学生生活で利用できるWebシステムの設計書一式
 - 要件定義書、DBテーブル定義表、ER図で修正があれば再提出
 - 実行結果と考察をまとめたもの