フレームワーク応用実習

01. Javaとデータベースの連携

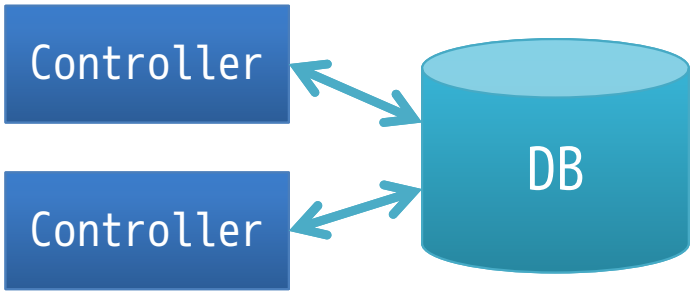
株式会社ジードライブ

今回学ぶこと

- DAOパターンとは
- JDBCとは
- Springプロジェクトのデータベース連携
- MyBatisの概要

DAOパターンとは

- DAO = **D**ata **A**ccess **O**bject
- データアクセスの実装のみを別のクラスに分離する設計パターン
  - データベースの変更やデータ保存方式の変更などが容易になる



DAO無し

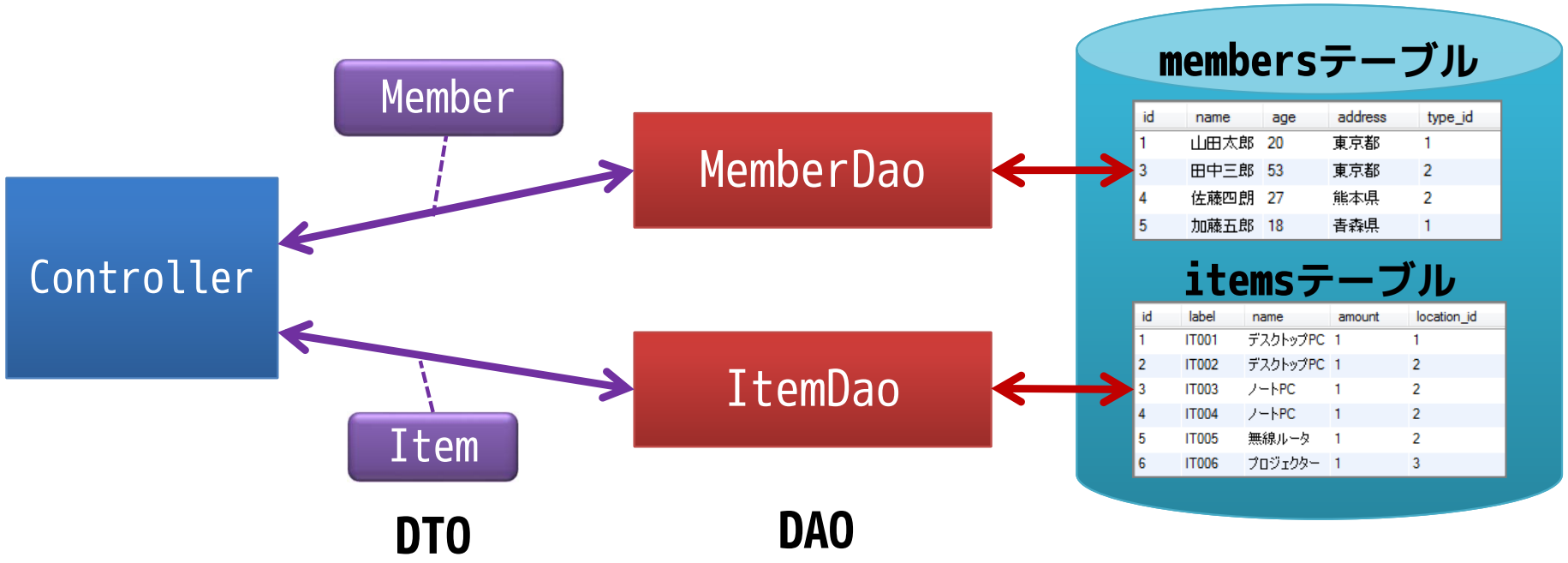
DAOパターン

DAOパターンの実装方法

- DAOパターンの実装方法には様々なバリエーションがあるが、少なくとも以下の点は共通している
  - 1つのテーブルに対して1つのDAOクラスを定義する
  - ControllerとDAOとの情報交換用のクラス(DTO = Data Transfer Object)を定義する

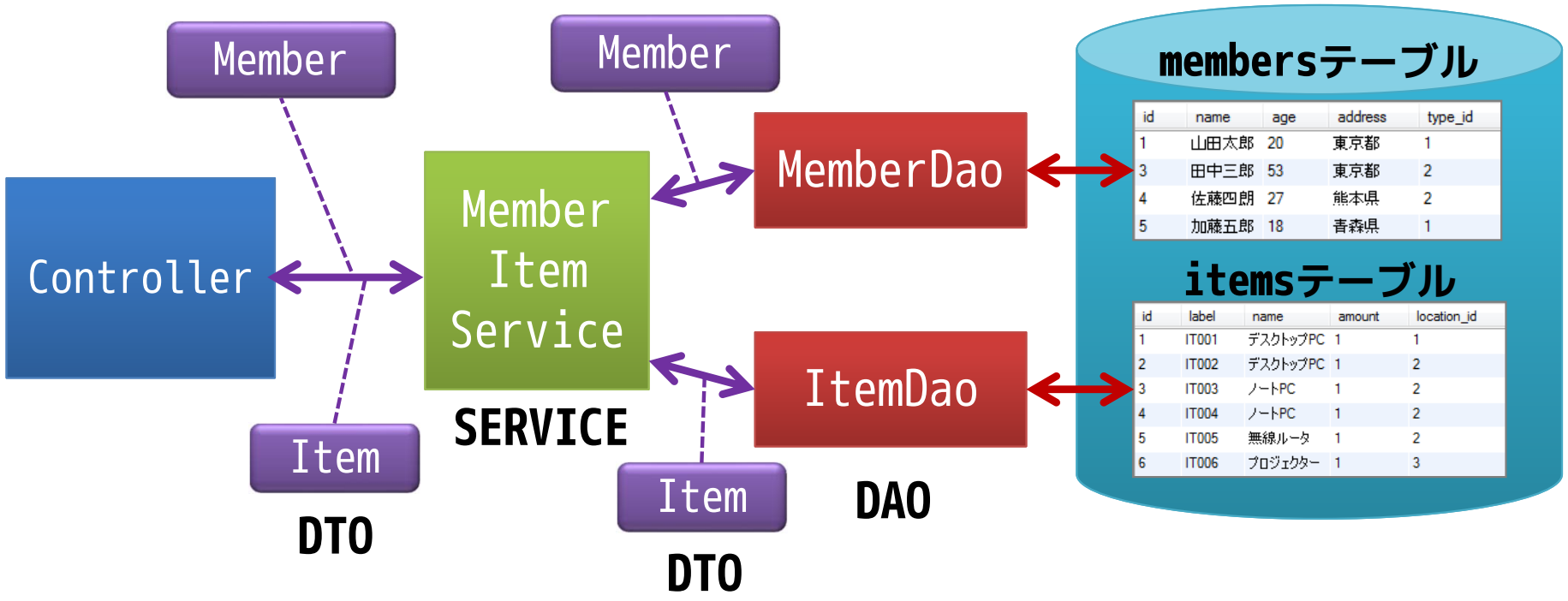
DAOパターンの利用イメージ(1)

- コントローラーから直接DAOクラスを利用する
  - 一度の処理で単一のテーブルにのみアクセスする単純なシステムの場合に有効



DAOパターンの利用イメージ(2)

- サービスと呼ばれる仲介役クラスを作り、コントローラーからはサービスを呼び出す形にする
  - 一度の処理で複数のテーブルにアクセスするシステムの場合に有効



DTO(Data Transfer Object)の例

- DTOクラスは、DBのテーブルのデータを表す

Memberクラス

```
public class Member {
  private Integer id;
  private String name;
  private Integer age;

  // 各フィールドのアクセッサ
  public Integer getId() {
    return id;
  }
  public void setId(Integer id) {
    this.id = id;
  }
  …
}
```

1行分のデータを
格納するフィールド

membersテーブル

id	name	age
1	山田太郎	27
2	鈴木次郎	31
3	加藤三郎	29

DAO(Data Access Object)の例

MemberDaoインターフェース

```
public interface MemberDao {
  List<Member> findAll() throws Exception;
  Member findById(Integer id) throws Exception;
  void insert(Member member) throws Exception;
  void update(Member member) throws Exception;
  void delete(Member member) throws Exception;
  …
}
```

データの選択、追加、
更新、削除メソッド

実装

MemberDaoImplクラス

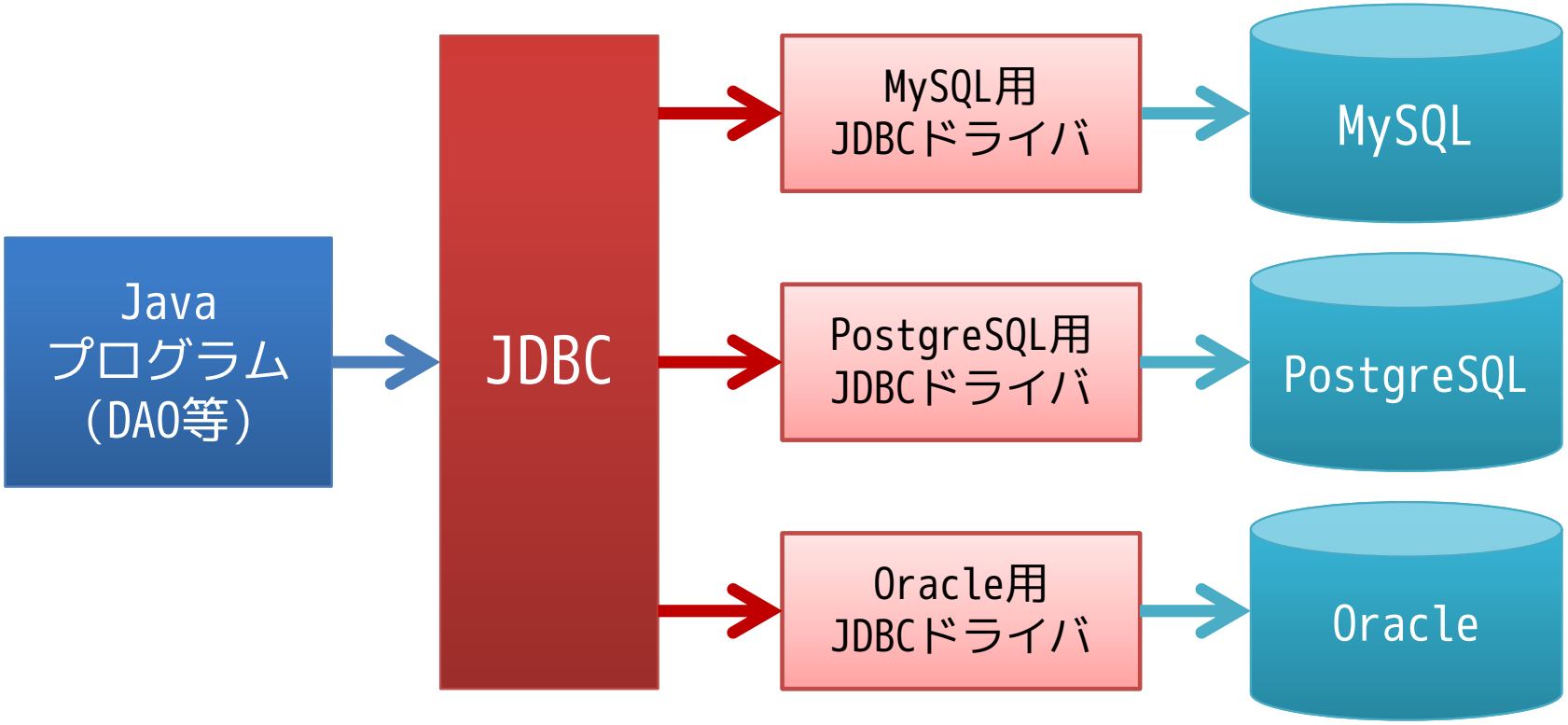
```
public class MemberDaoImpl implements MemberDao {
  @Override
  public List<Member> findAll() throws Exception {
    List<Member> memberList = new ArrayList<>();
    …
    return memberList;
  }
  …
```

JDBCとは

- **Java DataBase Connectivity**
- Javaプログラムからデータベースを操作するためのAPI (仕様)
  - JDBCに準拠した**JDBCドライバ**を利用することで、Javaプログラムから統一的な処理手順で様々なデータベースを利用することができる
- データベースごとにJDBCドライバが存在する
  - MySQL用JDBCドライバ
  - PostgreSQL用JDBCドライバ
  - Oracle Database用JDBCドライバ、等

JDBCの役割

- データベースごとの違いを吸収し統一的なデータベース操作インターフェースを提供する



Springプロジェクトとデータベース

- JDBCを直接利用するプログラムは煩雑になることが多い
  - try-catch文やSELECT文で取得した結果をJavaのオブジェクトにマッピングする記述が煩雑になる
  - データベース製品によって例外の型が異なるためデータベース製品変更時の改修が必要になる

Springの導入によって解決できる

SpringがJDBCとの仲介役となって、煩雑な処理を担う

データベースとの連携

- Springと以下のようなフレームワークを組み合わせて、データベースとの連携を行う

フレームワーク	説明
MyBatis	SQLをメソッドに紐づけて、データベース操作を行う SQLは、XMLまたはアノテーションで記述する
Spring Data JPA	SQLを記述することなく、メソッドでデータベース操作を行うことができる。必要に応じてSQLを使用することも可能
JOOQ	Javaのプログラムから SQL文を生成する データベース製品ごとの方言の差異を吸収してくれる
Flyway	データベースのマイグレーションツール ⇒ データベースの状態をバージョン管理できる

WebシステムへのMyBatis導入

- フレームワーク応用実習では、 WebシステムにMyBatisを導入し、データベースとの連携を図る