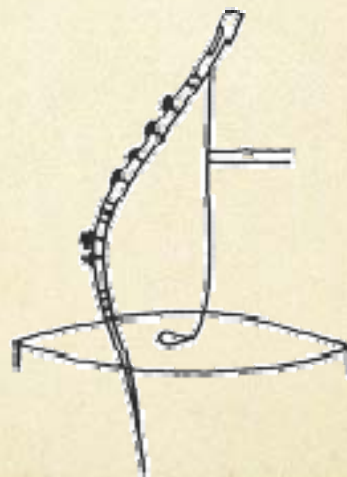


DBFlute はどうなの？



DBFluteとは？

現場指向の
O/Rマップ

1. 完全タイプセーフなSQL自動生成

2. 改良された2WaySQL

3. 現場重視の機能を装備

ConditionBean

1. 完全タイプセーフなSQL自動生成

1

テーブル名・カラム名
を完全補完

- Eclipseで「.(ドット)」補完
で安全実装
- DB変更に強い(コンパイル
エラーで検知)

2

目的ベースの
インターフェース

- ヒューマンミス等の
SQLExceptionは発生しない

【DBFluteの考え】

プログラム上でSQLを
組み立てる時の必須要件

3

外だしSQLとの
明確な線引き

- やれることが限られている
- 方針がなくてもバラバラ
実装にはなりにくい

【DBFluteの考え】

想定外コストはデバッグ作業で発生する。

▼
安全性を重視したインターフェースが、結果的にコストを削減する。

外だしSQL(OutsideSql)

2. 改良された2WaySQL

1

エラーメッセージ
を徹底

- タイプセーフでないので
デバッグのためにとても重要

2

戻り値クラス
を自動生成

- SQLから戻り値クラス
を自動生成
- マッピングミス発生しない
- SQLの文法チェックになる

3

SQLファイル指定が
タイプセーフ

- SQLファイルから
パス定義を自動生成
- SQLファイルがないエラー
が発生しない

現場指向機能

3. 現場重視の機能を装備

共通カラムの自動設定

区分値の解決

ページング検索を徹底サポート

DDLの実行・テストデータの登録の自動
(ReplaceSchema)

デバッグログの見易さ重視 etc...

これらは皆、現場から生まれてきた機能である。
(作者自身も現場でプログラマとしてDBFluteを利用)

様々なプラットフォーム

ニーズの高いその他の
プラットフォームに対応

Seasarだけでなく



Spring Framework
でも動作

Javaだけでなく



.NET-3.0(C#)
でも動作
(C#のクラスを生成)

プロジェクト間でリソース(人員)の
入れ替えをした場合の学習コスト削減