

# デザインパターン概説

2025.02.16 金田篤実

# 目次

- デザインパターンとは
- デザインパターンを学ぶことのメリット
- Factoryパターン

# デザインパターンとは

- 過去のソフトウェア設計者が構築した設計ノウハウをパターン(型)にしてまとめたもの。
- デザインパターンはオブジェクト指向を有効に使うために作られたものである。
- デザインパターンとしてまとめられているものにもいくつか種類がある。最も有名なものは"GoF"と呼ばれるデザインパターンである。

# デザインパターンを学ぶとどうなる？

- デザインパターンにはオブジェクト指向言語をどんな感じでうまく使ってプログラミングしていくかというノウハウが詰まっているため、デザインパターンを学ぶ＝オブジェクト指向の有効な使い方を学ぶということになる。
- パターンをより多く知っておくことで、ソフトウェア設計の現場で多くのパターンの中から最も有効なものを選択できるようになる。

# Factoryパターン

インスタンス生成を一元管理し、コードの柔軟性と再利用性を高めるためのデザインパターンである。主に下記の3パターンがある。

- **Simple Factoryパターン :**

シンプルでわかりやすいが、大規模なソフトウェアでは他の2つを使用する方が良い。

- **Abstract Factoryパターン :**

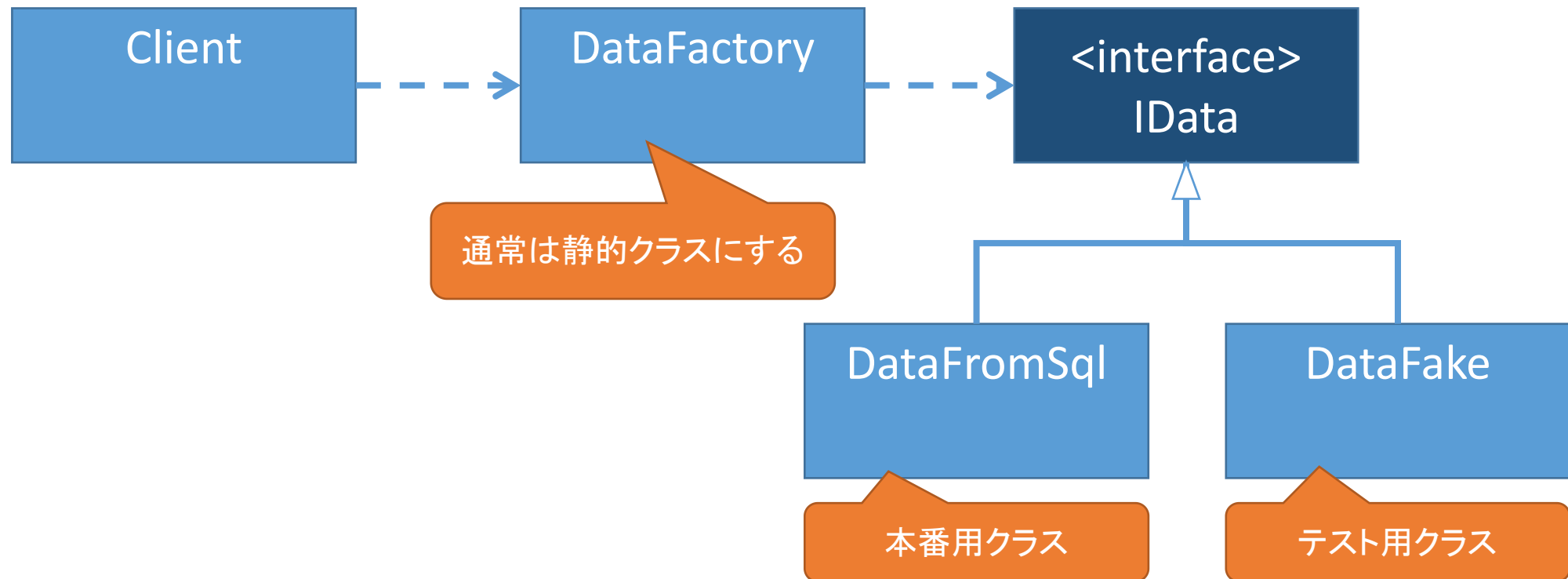
Simple Factoryをグループ化したようなイメージ。

- **Factory Methodパターン :**

少し複雑なので今回は説明省略

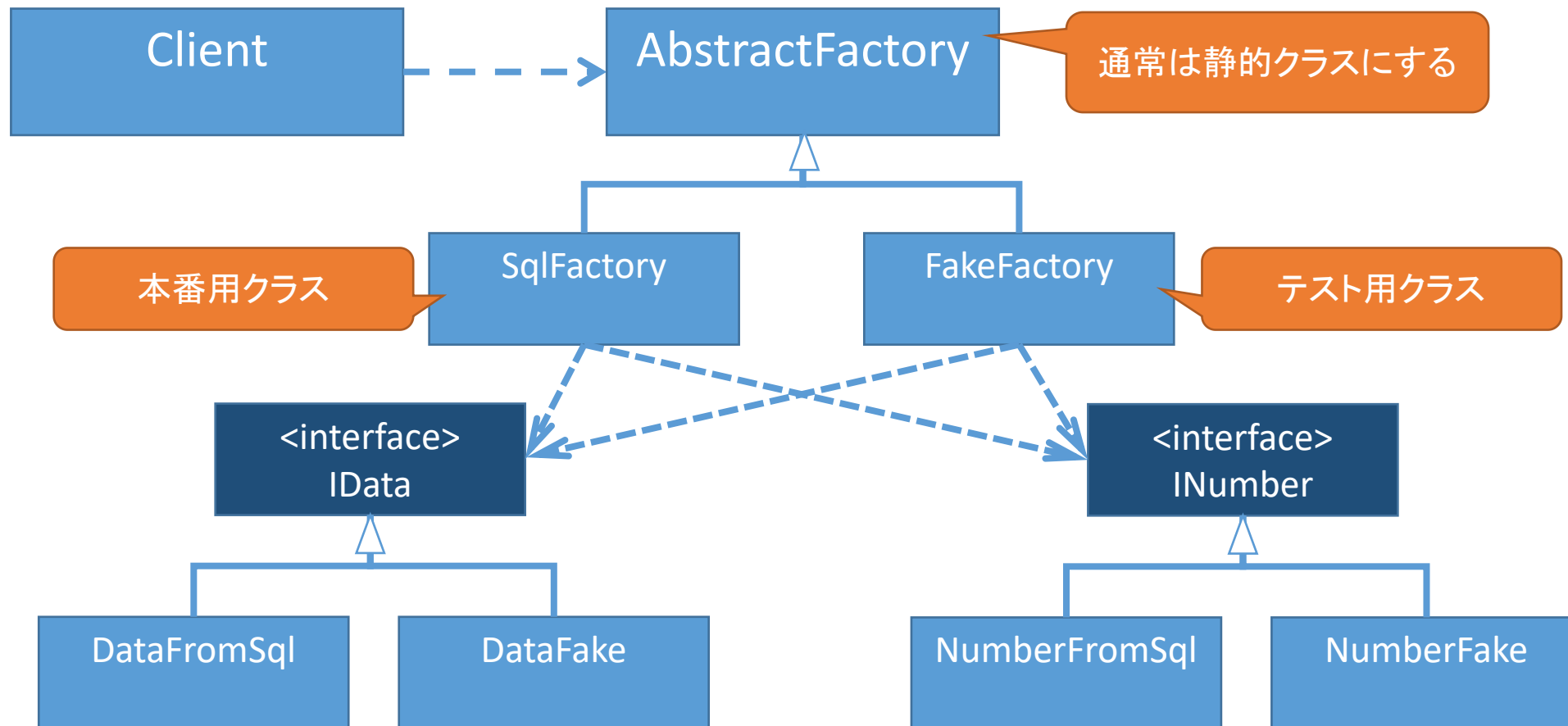
# Factoryパターン

## Simple Factoryパターンのクラス図



# Factoryパターン

## Abstract Factoryパターンのクラス図



# NullObjectパターン

クライアントが毎回オブジェクトのNullチェックをしなくても良いようにサービス側はNullの代わりにNullオブジェクトを返却する。

## NullObjectパターンのクラス図

