Data access object（DAO）

Data access object
https://en.wikipedia.org/wiki/Data_access_object
shikimi 2017-03-20 13:35:49

Data Access Object（DAO）
*この記事はソフトウェア・デザインパターンについてのものである。マイクロソフトのライブラリについては*Jet Data Access Objects*を参照のこと。*
コンピュータ・ソフトウェアにおいて、データ・アクセス・オブジェクト（DAO）とは、一種のデータベースや永続性機構の抽象的なインタフェースを提供するオブジェクトである。永続化層を呼び出すアプリケーションをマッピングすることで、DAOはデータベースの詳細を明らかにすることなく特定のデータの操作を可能にする。この分離は単一責任の原則を支える。それは、ドメイン固有のオブジェクトとデータ型の点からアプリケーションがどのようなデータアクセスを必要とするか（DAOのパブリック・インタフェース）を、それらの要求が特定のDBMSやデータベース・スキーマなどでどのようにして満たされることが出来るか（DAOの実装）から分離する。
このデザインパターンは以下のものに一様に適用可能である：（1 - ほとんどのプログラミング言語；2 - 永続性を必要とするほとんどの種類のソフトウェア；3 - ほとんどの種類のデータベース）。だが、伝統的にJava EE applicationと関係データベース（そのプラットフォームに関してサン・マイクロシステムズのベスト・プラクティス・ガイドライン［1］『Core J2EE Patterns』をもとにしているため、JDBC APIを介してアクセスされる）と結び付けられる。
目次

利点
データ・アクセス・オブジェクトを利用する利点は、互いについて知ることはできるが知るべきではなく、頻繁に独立して展開することが予期されているアプリケーションの重要な2つの部分を、比較的単純かつ厳密に分離できることである。ビジネス・ロジックの変更は同じDAOインタフェースに依存することがあるが、永続ロジックへの変更はインタフェースが適切に実装されている限りDAO clientに影響しない。格納されたものの詳細はすべてアプリケーションの他の部分からは隠される（カプセル化を参照）。このようにして、一つのDAOの実装を修正するだけで、残りのアプリケーションが影響を受けることなく、永続性機構に行われうる変更を実行することが可能となる。DAOはアプリケーションとデータベースとの間の橋渡し役をする。それらはオブジェクトとデータベースのレコードとでデータを行ったり来たりさせる。コードの単体テストは、試験のテストダブルとDAOを置き換えることで容易になり、それによってテストを永続化層に依存しないようにする。
Java プログラミング言語の非特異的背景において、設計思想としてのデータ・アクセス・オブジェクトは様々な形で実現されうる。これはデータ・アクセス部分をアプリケーション・ロジックから分離する極めて単純なインタフェースから、フレームワークや市販用の製品にまで渡る。
DAOのコーディングの枠組みは多少の技術を必要とすることがある。Java Persistence APIやEnterprise JavaBeansのような技術はアプリケーション・サーバーに組み込まれるようになり、

JavaEEアプリケーション・サーバーを利用したアプリケーションで使われることが可能である。TopLinkのような市販用の製品はオブジェクト関係マッピング（ORM）に基づき利用可能である。一般的なオープンソースのORM製品には、DoctrineやHibernate、iBATIS、Apache OpenJPAのようなJPA implementationsなどがある。

欠点

DAOを使用することの潜在的な欠点には、漏れのある抽象化やコードの重複、抽象化の逆転がある。とりわけ、正規のJavaオブジェクトとしてのDAOの抽象化はそれぞれのデータベース・アクセスのコストの高さを隠すことがあり、また、それ以外の場合では通常のSQLの集合演算での単一の演算で返されるであろう情報を取得するために、開発者に倍量のデータベースクエリを起動させることを強いることになる。あるアプリケーションが複数のDAOを必要とする場合、それぞれのDAOに対して本質的に同じ生成と読み取り、更新、削除のコードを繰り返していることに気付くかもしれない。しかしながら、この定型文のコードは、これらの共通の操作を取り扱う汎用のDAOを実装することで避けられるかもしれない［2］。時間消費が抑えられる。

ツールとフレームワーク

ODB　C++用のコンパイラベースのオブジェクト関係マッピング（ORM）システム

ORMLite　JDBCとAndroid用のJavaの軽量オブジェクト関係マッピング（ORM）フレームワーク［3］

Microsoft Entity Framework

DBIx::Class　Perl用のオブジェクト関係マッピング（ORM）モジュール

関連項目

Create, read, update and delete (CRUD)

Data access layer

Service Data Objects

参考文献

1. "Core J2EE Patterns - Data Access Objects". Sun Microsystems Inc. 2007-08-02.

2. 参照： http://www.ibm.com/developerworks/java/library/j-genericdao/index.html for workarounds

3. Hodgson, Kyle; Reid, Darren. *ServiceStack 4 Cookbook*. Packt Publishing Ltd. p. Chapter 4. ISBN 9781783986576 . Retrieved 22 June 2016.

外部リンク

Java Persistence - The DAO Design Pattern

PHP best practices (Use Data Access Objects (DAO))
