

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Design Patterns

by Versus 2020

Data Access Object **Объект доступа к данным**

Структурные шаблон





Суть шаблона

Прослойка между СУБД и бизнес слоем.



Проблема

Данные нужно получать и сохранять информацию в нескольких источниках.

API доступа к персистентному хранилищу данных может зависеть от поставщика продукта.

Специфичность механизмов доступа и API прямо влияет на переносимость компонентов.

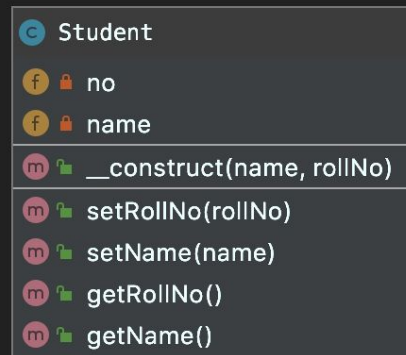
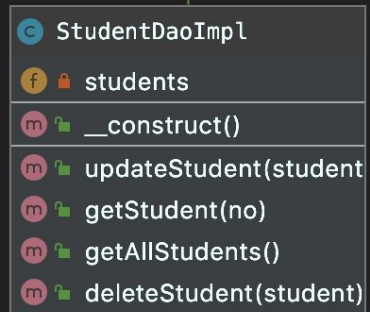
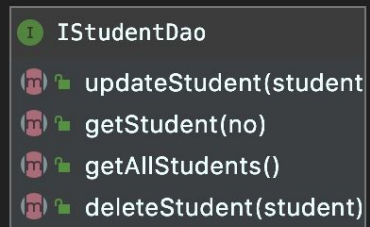
Должна быть обеспечена легкая миграция на других поставщиков данных, тип хранилища.



Решение

Абстрагировать и инкапсулировать доступы к источникам данных.

Структура



Пример кода





Применимость

Единый интерфейс к доступу слою данных, независимо от хранилища, API и механизма хранения.



Шаги реализации

1. Определяем объект данных
2. Определяем общий интерфейс работы со слоем данных.
3. Определяем класс доступа к данным, реализующий интерфейс.
Методы должны реализовать логику работы со слоем данных.
4. Клиент работает непосредственно с объектом доступа к данным.



Преимущества

Инкапсуляция механизмов работы с данными, типов хранилища, слоя данных.

Переносимость между информационными потоками.

Полиморфизм, когда несколько классов реализуют обобщенный интерфейс, а сервис, инкапсулирующий в себе бизнес-логику, работает только с этим интерфейсом.

Облегчает модульное тестирование бизнес-слоя.



Недостатки

Паттерн DAO предоставляет размытое описание контракта.

Поддержка конструкции "интерфейс сервиса - сервис - интерфейс DAO - одна или несколько реализаций DAO" довольно утомительна.

ИТОГ

