

Lista zaawansowanych wzorców projektowych

1. Data Mapper – umożliwia wymianę danych pomiędzy obiektami a bazą danych. Używany przez bibliotekę Hibernate. Przykładowe klasy: klasy oznaczone jako @Entity (Station.java, CurrentPrice.java, etc). Łatwe zastąpienie warstwy realizowanej przez Data Mapper, możliwość niezależnego rozwijania, modyfikowania, testowania.
2. Data Transfer Object – umożliwia przenoszenie dane między procesami i zmniejszenie liczby wywoływanych metod. Przykładowe klasy: CurrentPriceDto.java, PriceRecordDto.java. Wzorec pozwala na ograniczanie i grupowanie danych.
3. Service Layer – warstwa określająca zbiór dostępnych operacji. Przykładowe klasy: PriceService.java, StationService. Service Layer hermetyzuje logikę biznesową umożliwiając sterowanie transakcjami i niwelując powielanie kodu.
4. Metadata Mapping – przechowywanie informacji dotyczących odwzorowań obiektów na dane relacyjne. Przykładowe klasy: CurrentPrice.java, PriceRecord.java, Station.java. Metadanymi są adnotacje Javy, np. @Id, @GeneratedValue, @ManyToOne, @Enumerated, @Entity. Metadata Mapping pozwala zdefiniować odwzorowania w krótki i przejrzysty sposób.
5. Repository – łączy warstwę dziedziny i odwzorowania danych. Przykładowe klasy: PriceRecordRepository.java, StationRepository.java, CurrentPriceRepository.java. Repository pozwala aplikacji na komunikację z bazą danych.
6. Value Object – równość obiektów zależy od równości pól, a nie referencji. Przykładowe klasy: wszystkie klasy z przeładowanymi metodami Equals() i GetHashCode(), na przykład Price, CurrentPrice, CurrentPriceDto, itd.. Value Object jest przydatny szczególnie w Javie, kiedy chcemy być w stanie porównywać ze sobą obiekty w inny sposób niż przez referencję. Szczególny przypadek – Money – obiekt reprezentujący wartość monetarną. Klasa: Price.java. Money pomaga zapisać obiekt reprezentujący pewną wartość pieniężną.
7. Identity Field – przechowuje pole identyfikacyjne z bazy danych. Przykład: PriceRecord.java pole id. Umożliwia jednoznaczną identyfikację rekordu w bazie danych.
8. Foreign Key Mapping – odwzorowuje asocjacje między tabelami jako klucz obcy tabeli. Przykład: PriceRecord.java pole station z adnotacją @ManyToOne.
9. Dependent Mapping – szczególny przypadek – Embedded Value – odwzorowuje obiekt do kilku pól tabeli innego obiektu. Przykład: CurrentPrice.java/PriceRecord.java oraz Price.java.