

УДК 004.4

В.М. Юзьвак

Тернопільський національний технічний університет імені Івана Пулюя, Україна

ПАТЕРНИ РОБОТИ З БАЗАМИ ДАНИХ: ООП-ПІДХІД

V.M. Iuzvak

DATABASE DESIGN PATTERNS: OOP APPROACH

Обрання прийомів зчитування та запису у бази даних є одним з наріжних каменів при проектуванні будь-якої інформаційної системи (ІС). В рамках ООП можна виокремити наступні підходи до зчитування і запису інформації у реляційні бази даних (патерни): 1) примітивний підхід з використанням PDO/масивів, 2) ActiveRecord, 3) DataMapper, 4) TableDataGateway.

Примітивний підхід не передбачає створення жодних класів, оскільки за допомогою PDO чи аналога дані вивантажуються безпосередньо у відповідний масив. Підходить для використання у високонавантажених проектах (за рахунок економності добування інформації), а також у невеликих проектах, де звернення до бази даних у сирцевому коді зустрічається нечасто. Недоліком даного підходу є необхідність щоразу писати SQL-запити, а також складність координації роботи кількох розробників.

Інші підходи використовують сутності – клас, який відображає один запис у таблиці. Використання такої абстракції дозволяє приблизити підходи в роботі з реляційними базами даних до роботи з класами в ООП і дозволяє відійти від постійного написання SQL-запитів. Код, який реалізує завантаження даних з бази у сутність і збереження сутності у базу даних називають ORM (Object-Relational Mapper).

ActiveRecord характеризується тим, що методи завантаження інформації та збереження інформації у базу даних містяться у класі самої сутності. Також у сутності передбачається метод, який співставляє поля об'єкту сутності із полями у базі даних. Зазвичай, такі методи містяться у базовому класі з якого спадкують класи конкретних сутностей. Клас конкретної сутності може містити і додаткові методи, які обробляють прочитані з бази дані. Основним недоліком даного підходу є змішування бізнес-логіки і роботи з базою даних в межах одного класу, тобто порушення принципу SRP – принципу єдиного обов'язку класу, що входить до SOLID [1]. ActiveRecord як патерн знайшов реалізацію у фреймворку Yii 1 та Doctrine 1.

Недолік щодо порушення SRP відсутній при патерні DataMapper. При даному підході код завантаження даних з бази та запису даних у базу виокремлений в окремий клас, який не містить бізнес логіки. Даний патерн реалізований у класі Repository ORM Doctrine 2.

TableDataGateway як патерн нагадує DataMapper, але може бути реалізований без об'єктів-сутностей – для будь-яких сутностей використовується один і той же клас. Прикладом реалізації патерну є клас Zend_Db_Table_Row із фреймворку Zend.

Обрання необхідного патерну залежить від конкретних особливостей ІС – з одної сторони, використання патернів із ORM полегшують роботу з базою даних та зменшують кількість необхідного коду, а з іншої – використання ORM створює додаткове навантаження, що робить їх неприйнятними для високонавантажених систем.

Література

1.S.O.L.I.D: The First 5 Principles of Object Oriented Design [Електронний ресурс] / – Режим доступу: <https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design> – Назва з екрана.