Ханджанов Д.О., магістрант,
Єфіменко А.А., к.т.н., завідувач кафедри КІ та КБ,
Житомирський державний технологічний університет

ОСНОВНІ ПАТТЕРНИ РОБОТИ З РЕЛЯЦІЙНОЮ БАЗОЮ ДАНИХ В ВЕБ АРХІТЕКТУРІ

Сучасні вимоги до веб програм полягають у можливості зберігати, обробляти та надавати користувачу певну інформацію. Для цих цілей часто використовують реляційні СУБД на основі MySQL. Для роботи з такою БД в об'єктно-орієнтованій архітектурі існує 2 основних патерни: ActiveRecord (AR), DataMapper (DM). Ці патерни активно використовують сучасні фреймворки для полегшення та пришвидшення розробки програмного забезпечення. Для реалізації даних паттернів часто використовують технологію Doctrine.

Паттерн AR — це шаблон проектування, що використовується при реалізації доступу до реляційних баз даних. AR реалізує популярний підхід об'єктно-орієнтованого проекціювання (ORM). Кожен клас AR відображає таблицю бази даних, екземпляр AR — запис цієї таблиці, а загальні операції CRUD реалізовані як методи AR. Принцип роботи проілюстрований на рис. 1. В результаті можна працювати з об'єктом моделі достатньо простим та ефективним способом. Але в такому випадку модель включатиме в собі логіку обробки інформації та логіку CRUD одночасно, що в свою чергу порушує принципи SOLID, а саме принцип єдиної відповідальності. Систему яка застосовує даний патерн можна буде достатньо швидко та дешево створити, але вона матиме проблеми з розширенням та супроводженням. Даний спосіб підходить для невеликих систем, або для проектів які застосовують фреймворки що підтримують даний підхід та в яких реалізується нескладна логіка обробки даних.

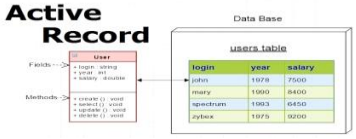

Рис. 1. Діаграма ActiveRecord

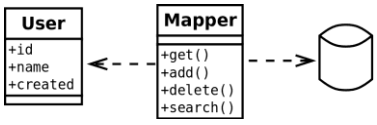

Рис. 2. Діаграма DataMapper

Паттерн DM – це патерн, який виступає в ролі посередника для двобічної передачі даних між постійним сховищем даних і представлення даних в пам'яті. Мета патерну в тому, щоб тримати уявлення даних в пам'яті і постійне сховище даних незалежними один від одного і від самого перетворювача даних. На відміну від AR даний патерн повністю відповідає принципам SOLID, тому що розділяє логіку обробки даних від логіки CRUD. Система створена на основі даного патерна буде добре розширюватись та підстроюватись під вимоги користувача (розділена логіка досить просто дозволить замінити модель, або мапер, замінивши тим самим логіку роботи моделі, або джерело даних), але також вона буде складнішою для зрозуміння та буде мати щонайменше вдвічі більше об'єктів з якими доведеться працювати. Даний патерн краще застосовувати в великих системах, або системах які реалізують складну логіку обробки даних.

Doctrine — об'єктно-реляційний проектор (ORM) для PHP 5.3.0+, який базується на шарі абстракції доступу до БД. Однією з ключових можливостей Doctrine є запис запитів до БД на власному об'єктно-орієнтованому діалекті SQL, званий DQL і базується на ідеях HQL.

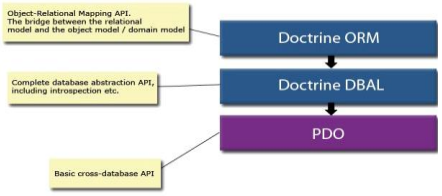

Рис. 3. ДіаграмаDoctrine

Існує дві версії даного проектору. Перша реалізує логіку роботи патерна AR тому програмісту для використання необхідно лише створити модель, подати дані та зберегти її методами одного класу. Друга версія більше опирається на логіку роботи патерна DM і програмісту при використані доведеться використовувати декілька об'єктів. Серед переваг даної технології можна відмітити: з 2006 року має дуже стабільну, якісною кодову базу; надзвичайно гнучкі та потужні об'єкт-карти та функції запиту; підтримка програмного забезпечення БД та низького рівня для всіх ваших випадків використання; Велика спільнота та інтеграція з багатьма різними системами (Symfony, ZendFramework, CodeIgniter, Flow, Lithium та ін.)

Основною відмінністю від класичного шляху реалізації паттернів є те що CRUD реалізується не за допомогою звичайних MySQL-запитів, а за допомогою DQL, який описується в моделі як коментар до метода або поля. Такий підхід досить зручний в використані та дозволяє використовувати багато вбудованих можливостей, але має свій поріг входження, підтримується лише з версії PHP 5.3.0 та потребує встановлення додаткових бібліотек.