УДК 004.4

А. К. Рижков
О. В. Войцеховська

ВИКОРИСТАННЯ ПАТЕРНУ «REPOSITORY» ПРИ ПРОЕКТУВАННІ РОЗПОДІЛЕНОГО СЕРВЕРНОГО ДОДАТКУ

Вінницький національний технічний університет

Анотація
В роботі проведено аналіз використання патерну «Repository» у проекті розподіленого слабкозв'язаного серверного додатку. Патерн використовується для зменшення зв'язності між рівнем бізнес логіки та рівнем бази даних.
Ключові слова: патерн, Repository, Entity framework.

Abstract
The paper analyzes the use of the "Repository" template in the projects of a distributed loosely coupled server application. The template is used to reduce the connection between the level of business logic and the level of the database.
Keywords: pattern, Repository, Entity framework.

Вступ

Патерн «Repository» дозволяє абстрагуватись від конкретних підключень до джерела інформації, з якими працює програма та є проміжним етапом між класами, які безпосередньо взаємодіють із базою даних та іншою програмою.

Серверна частина розроблюваної інформаційної системи взаємодіє із базою даних MS SQL Server [1]. Замість того, щоб безпосередньо працювати із контекстом даних, який відповідає за маніпуляції над даними, клас контролеру працює із класом репозиторію, у якому описана вся основна логіка для маніпуляції над даними, перед збереженням їх у базу даних.

Результати досліджень

Кожен репозиторій у проекті реалізує інтерфейс IRepository, у якому описані основні методи, які повинні бути описані у класі нащадку. У інтерфейсі описано методи: Create, Delete, GetById, Get, GetPagedItems, Update. Репозиторії, що реалізують цей інтерфейс, містять логіку для збереження, видалення, редагування, та отримання інформації про певну сутність. Так, наприклад, репозиторій який працює із новинами, буде зберігати інформацію про новини у базу даних. Приклад схеми зв'язків між контролерами та репозиторіями подано на рисунку 1.

Використання патерну «Repository» покращує гнучкість коду. Так, у випадку, якщо змінилась логіка збереження об'єкту у базу даних, зміни торкнуться лише класу репозиторію, а клас контролеру продовжить працювати із моделями запиту та відповіді [2].

Окрім цього, код, який відповідає за збереження інформації у базу даних, може бути повторно використаний. Це важливий принцип чистого коду – код, який може бути повторно використаний, повинен бути винесений у окрему функцію. Тому код буде легко зрозуміти новим розробникам, які будуть працювати над проектом в подальшому.

Коли вся логіка для роботи із базою даних централізована в одному класі – репозиторії, легко додавати нові функціональні можливості, або змінювати логіку у випадку, якщо база даних змінилась або потрібно ввести нову логіку для обробки інформації. При зміні системи управління базами даних розробнику рівня доступу до даних достатньо внести зміни у клас репозиторію для обробки інформації.

У проекті описаний загальний інтерфейс типу IRepostory. При реалізації він типізується класом сутності, із якою буде вестись робота у базі даних. Так, наприклад, для створення репозиторію, який працює із сутністю новин, при реалізації інтерфейсу описується типізація IRepository<News> [3]. Це означає, що для збереження або оновлення інформації цей репозиторій буде приймати об'єкт типу News для збереження цього об'єкту у базу даних.

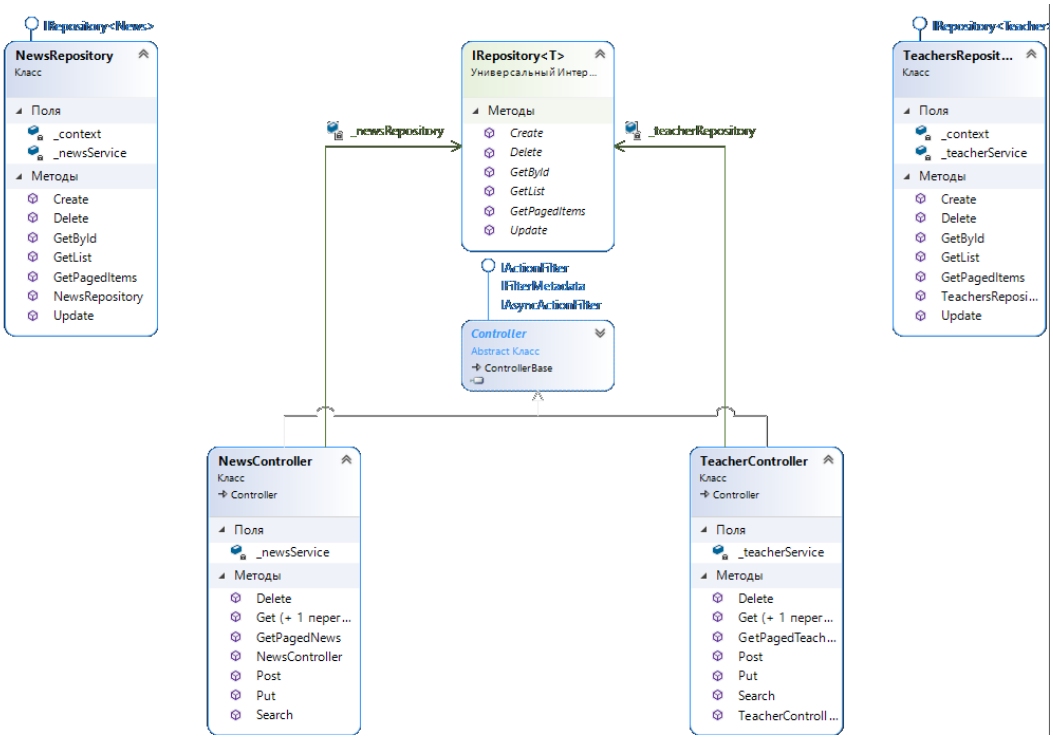

Рисунок 1 – Схема зв'язків між класами контролерів та репозиторіїв у проекті розподіленого слабкозв'язаного серверного додатку

Окрім цього репозиторій виконує важливу функцію – інкапсулює логіку для роботи із системою управління базами даних, що є основним принципом у об'єктно орієнтованому програмуванні [4]. Користувач не знає, як саме репозиторій взаємодіє із базою даних, але він може бути впевнений, що при створенні репозиторію типу IRepository<News>, він отримає об'єкт який за потреби додасть, видалить, відредагує або отримає інформацію про новини, що зберігаються у базі даних.

Актуальна технологія Entity Framework [5] дає можливість маніпулювати інформацією з бази даних безпосередньо через клас, описаний у середовищі .NET, приховуючи реалізацію від користувача.

Висновки

Отже, для забезпечення гнучкості та чистоти коду при проектуванні інформаційної системи потрібно розділяти різні функціональні обов'язки між різними класами. Для цього застосовано шаблон «Repository», який інкапсулює у собі логіку для роботи зі сховищем даних, що використовується у проекті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Аналіз архітектури розподіленого слабкозв'язного серверного додатку інформаційної системи / О. В. Войцеховська, А. К. Рижков // //Матеріали LI наукової-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ), 2022 р. [Електронний ресурс] Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15103/12731.

2. How to Implement the Repository Pattern in ASP.NET MVC Application [Електронний ресурс] – Режим доступу до ресурсу: https://www.infragistics.com/community/blogs/b/dhananjay_kumar/posts/how-to-implement-the-repository-pattern-in-asp-net-mvc-application

3. Repository pattern C# [Електронний ресурс] – Режим доступу до ресурсу: https://codewithshadman.com/repository-pattern-csharp/

4. Data and Encapsulation in complex C# [Електронний ресурс] – Режим доступу до ресурсу: https://www.dotnetcurry.com/patterns-practices/1367/data-encapsulation-large-csharp-applications

5. Overview of Entity Framework [Електронний ресурс] – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/ef/core/

Рижков Андрій Костянтинович – студент групи 1КІ-20мс, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: ryzhkovanruha@gmail.com.

Войцеховська Олена Валеріївна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця

Ryzhkov Andrii K. – students, 1KI-20ms, Faculty of information Technologies and Computer Engeneering, Vinnytsa National Technical University, email: ryzhkovanruha@gmail.com.

Voytsekhovska Olena V. — PhD, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University