**Лабораторна робота 25.**

**Реалізація REST-API сервісу для роботи зі студентською відомістю.**

**Мета роботи:** закріпити навички щодо створення класів-сутностей, репозиторіїв та роботи із ними в контролерах.

**Задачі:**

* Створити класи сутностей для всіх зовнішній ключів таблиць «іспити» та «іспитові бали»
* Створити класи сутностей для іспитових балів та іспиту
* Створити клас-репозиторій та реалізувати додаткові методи
* Створити клас-контролер та методи для отримання даних, додавання, зміни та видалення інформації
* Реалізувати користувацький інтерфейс для роботи із контролером
* Виконати індивідуальне завдання

1. Почнемо із створення додаткових класів-сутностей для обслуговування посилань зовнішніх ключів. Ми вже маємо класи «студент» та «група». Але у них треба реалізувати методи для експорту об’єктів даного класу у json-формат (рис.1-2).

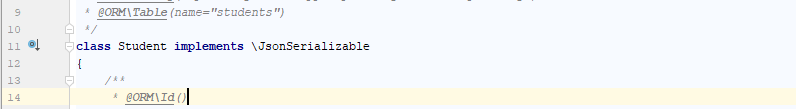




Рис.1. Доповнення у класі «студент»

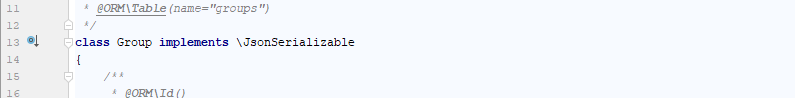
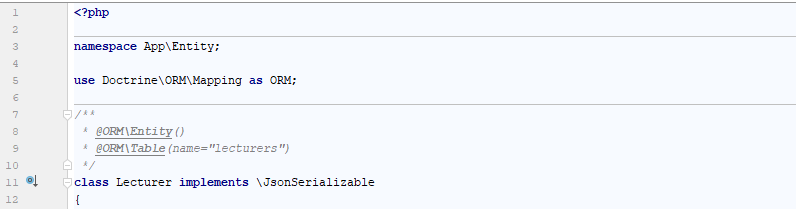
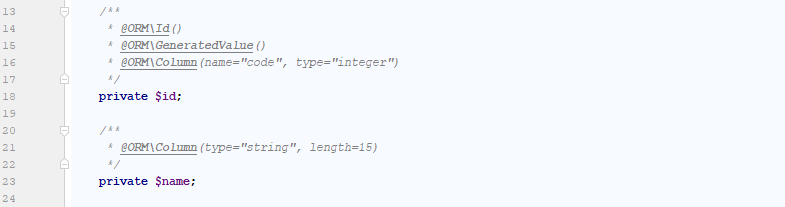


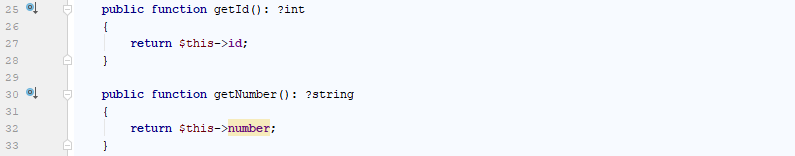


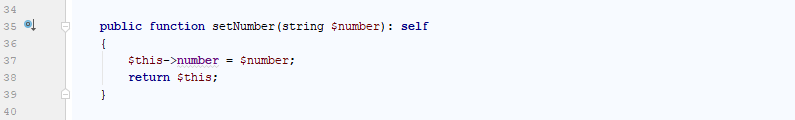
Рис.2. Доповнення у класі «група»

Також реалізуємо класи «викладач» (рис.3) та «дисципліна» (рис.4).









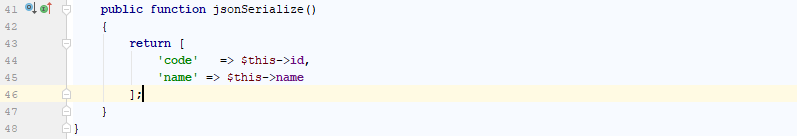
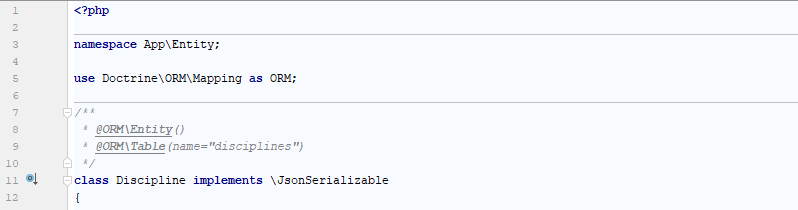
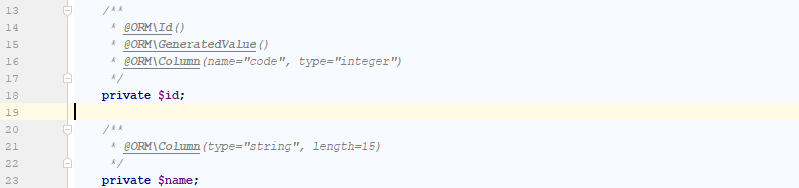
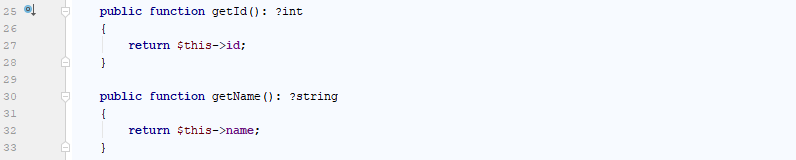
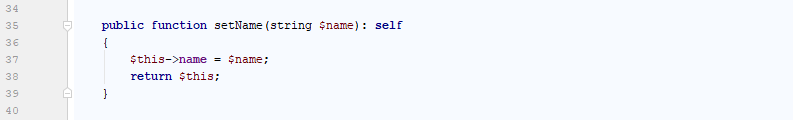


Рис.3. Клас «викладач»









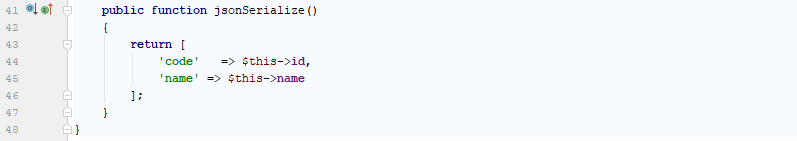
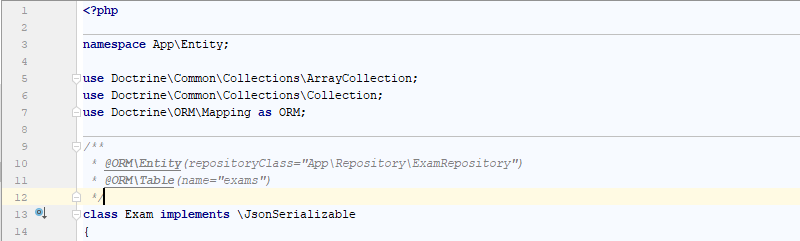
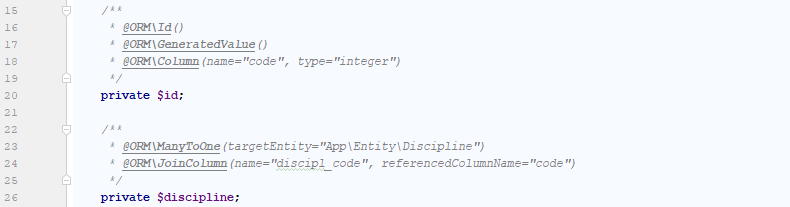
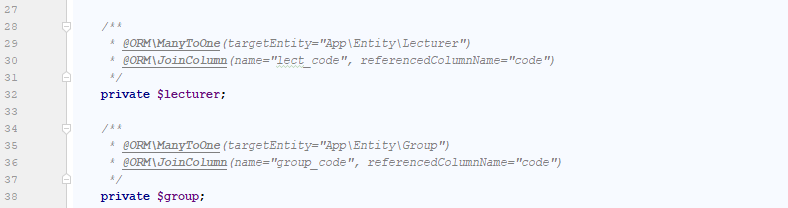


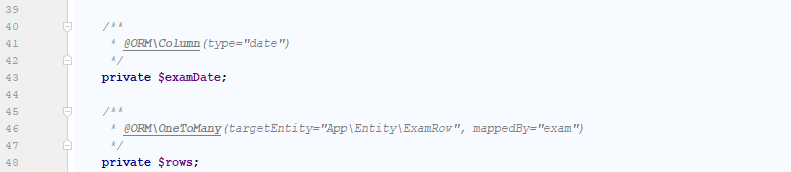
Рис.4. Клас «дисципліна»

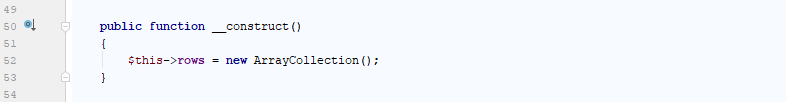
1. Далі реалізуємо клас «Іспит» із посиланнями та викладача, дисципліну та студентську групу. Клас також буде містити поле-колекцію «рядки», де буде представлено список студентів із балами (рис.5). Рядок даного поля буде посиланням на клас «іспитові бали», що буде реалізований після класу «Іспит».

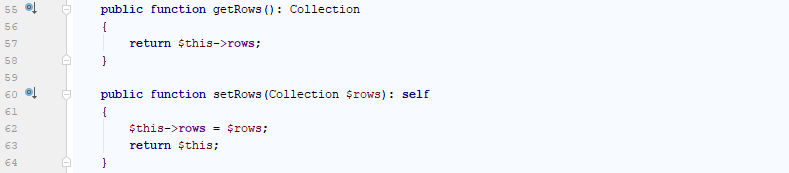




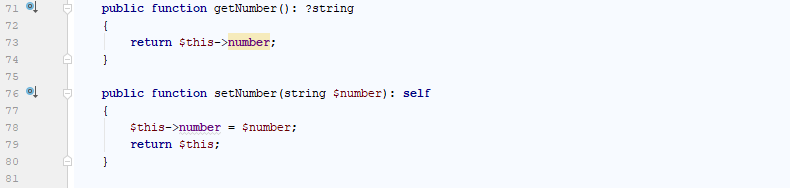


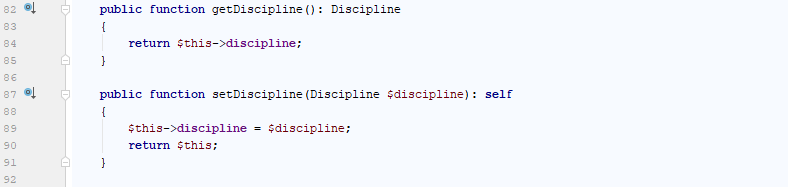


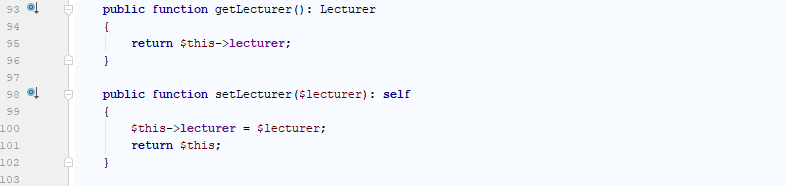


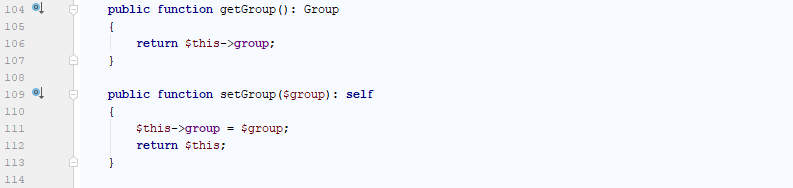


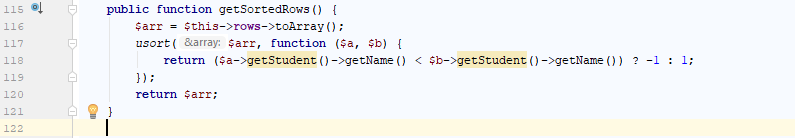












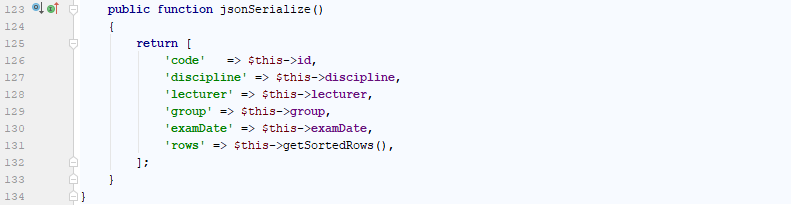
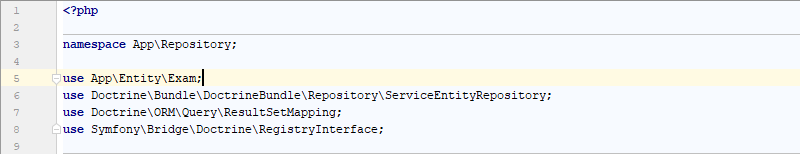
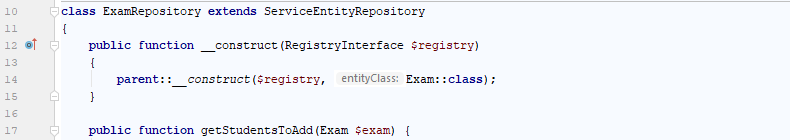


Рис.5. Клас «Іспит».

Клас «іспит» також буде мати клас-репозиторій (рис.6), у якому буде реалізовано метод для отримання списку студентів поточної групи, що ще не включені до відомості (кого можна додати).





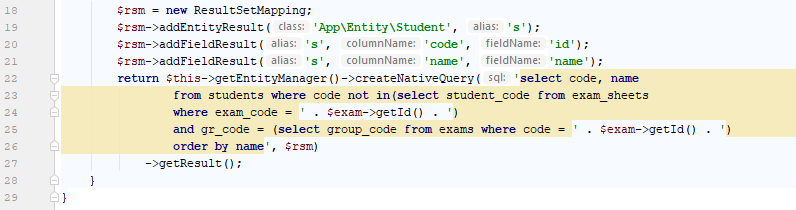
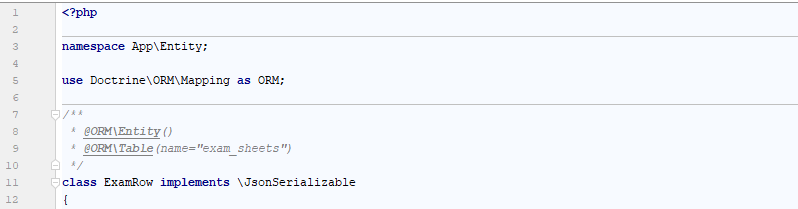
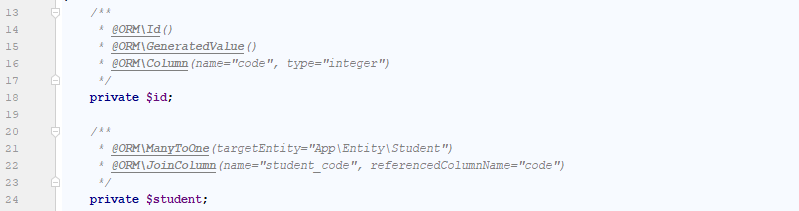


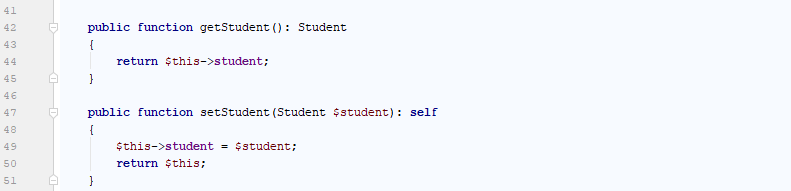
Рис.6. Клас-репозиторій для іспиту.

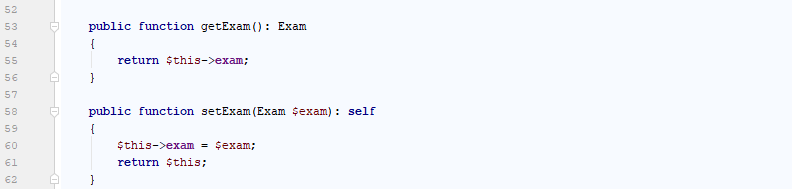
1. Реалізацію класу «іспитові бали» наводимо на рис.7.

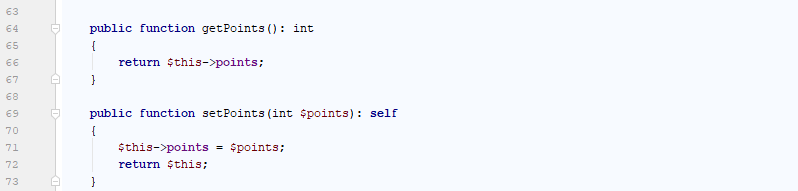












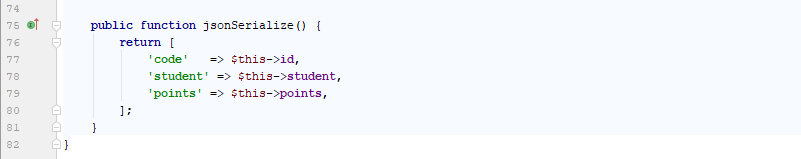
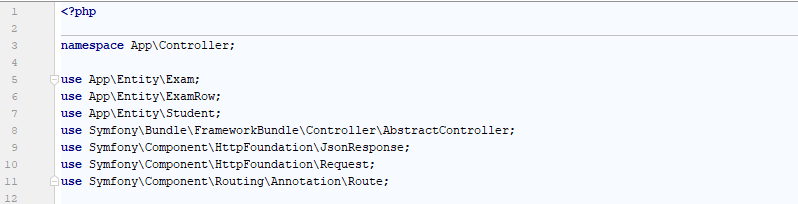
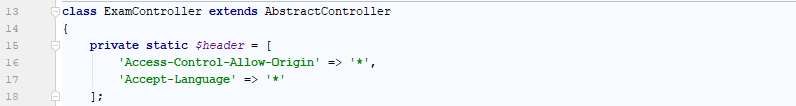
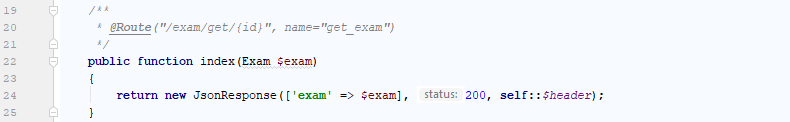


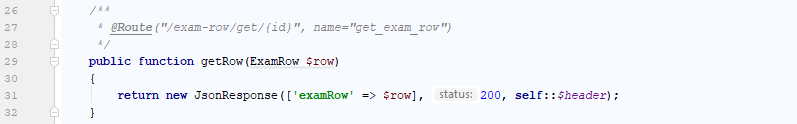
Рис.7. Клас «іспитові бали».

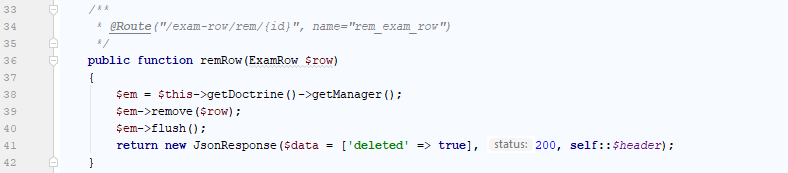
1. Виконаємо реалізацію класу-контролера, що оброблятиме запити клієнта, виконуватиме відповідні дії та повертатиме у разі необхідності результат у json-форматі (рис.8).

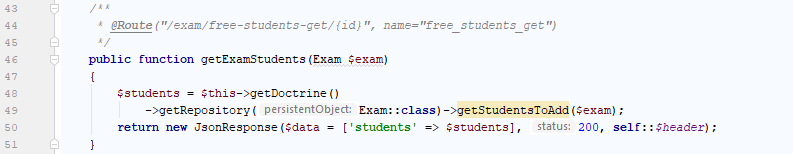


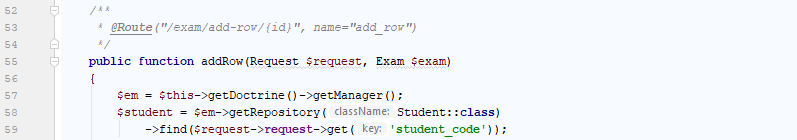


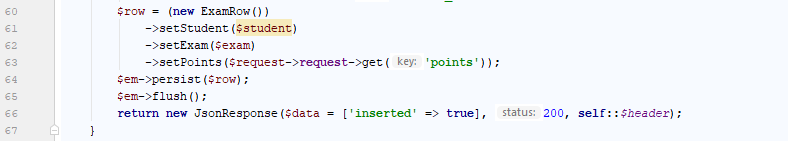












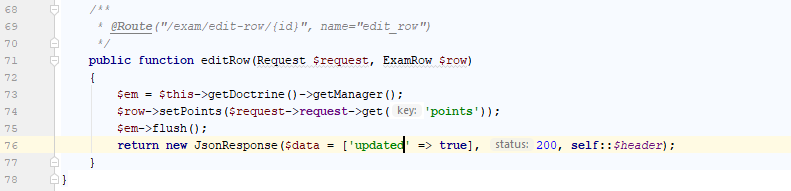
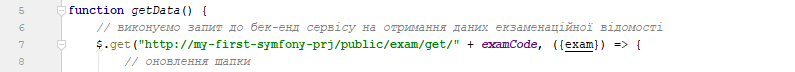
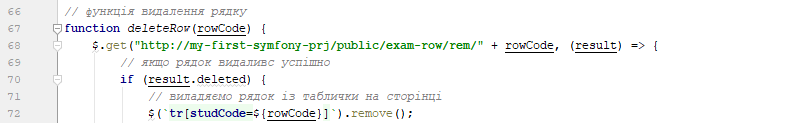
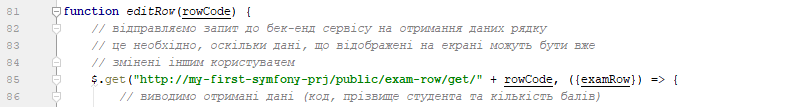


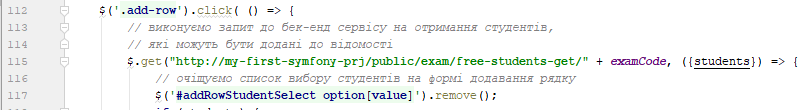
Рис.8. Контролер для роботи із студентською відомістю.

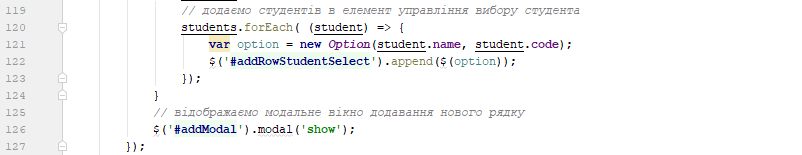
1. На цьому наш REST-API для роботи із студентською відомістю готовий. Далі реалізуємо користувацький інтерфейс для роботи із ним. Точніше ми використаємо вже реалізований у попередніх роботах (лаб.роб.17) інтерфейс, трохи змінивши запити та посилання для отримання json-даних. Створимо у нашому проекті папку «client» та скопіюємо туди файли index.html та папку «assets» із лабораторної роботи 17. Після цього виконаємо невеликі модифікації файлу «assets\js\main.js». На рис.9. наведені змінені рядки даного файлу.

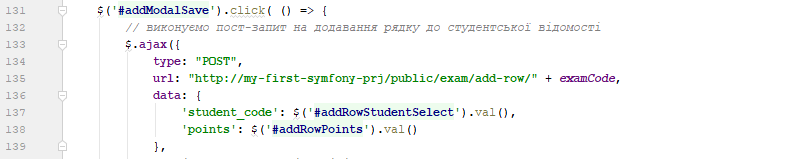












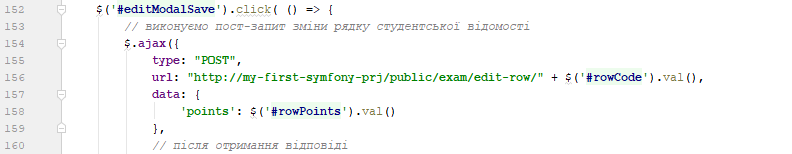


Рис.9. Зміни у «client\assets\js\main.js».

1. Перевіримо роботу користувацького інтерфейсу. Для цього у адресному рядку браузеру наберемо «http://my-first-symfony-prj/client/» (рис.10). Спробуємо додати, видалити та змінити рядок та пересвідчимось у коректрій роботі застосунку.



Рис.10. Зовнішній вигляд користувацького інтерфейсу.

1. Кінець роботи.

**Завдання для індивідуального виконання.**

Для власної БД реалізувати мінімум 2 класи сутностей для пов’язаних таблиць та реалізувати контролер для обслуговування операцій вибірки, додавання, зміни та видалення даних для обох таблиць.

.