

ПРАКТИЧНЕ ЗАНЯТТЯ № 09

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Тема: ASP.NET. Тришарова архітектура на базі ADO.NET та Dapper.

Мета: дослідження практичних аспектів реалізації веб-додатків на основі платформи ASP.NET на базі функціональних можливостей ADO.NET та DAPPER.

Програмне забезпечення: ОС Windows/Linux, Visual Studio, Visual Studio Code

Порядок виконання роботи:

1. Налаштування бази даних

- 1.1. У відповідності до сформованої теми роботи (технічне завдання) визначити сукупність таблиць (схему як структуру бази даних - <https://dic.academic.ru/dic.nsf/ruwiki/688432>) з якими буде відбуватись робота в розрізі проекту №1, в основі роботи якого буде використовуватись функціонал ADO.NET&DAPPER.
- 1.2. На рівні реляційної бази даних, в розрізі таблиць використовуваної схеми, передбачити наявність різного виду зв'язків між таблицями:
 - один до одного,
 - один до багатьох,
 - багато до багатьох.
- 1.3. Для однотипних CRUD запитів до бази даних передбачити наявність збережуваних процедур (4-8 шт.). Оцінити можливість реалізації Generic Repository на основі збережуваних процедур.

2. Робота із DAL. Розробка репозиторіїв

- 2.1. Використовуючи технологію доступу та керування даними ADO.NET розробити репозиторій із CRUD - запитам (1 *шт.* - чистий ADO.NET). Врахувати можливість роботи із збережуваними процедурами.
- 2.2. Використовуючи поєднання функціоналу ADO.NET та ORM-маппера Dapper розробити репозиторії із CRUD - запитам (2 *шт.* - ADO.NET & Dapper). Врахувати можливість роботи, за необхідності, із збережуваними процедурами.
- 2.3. При реалізації репозиторіїв на ADO.NET & DAPPER передбачити присутність:
 - параметричних запитів;
 - багатотабличних запитів (<https://sql-academy.org/ru/guide/multi-table-request-join>)

3. **Розробити функціонал шару бізнес логіки - BLL (сервіси/менеджери)**
Так, зокрема, для відповідного шару тришарової архітектури передбачити:
 - 3.1. Реалізацію функціональних вимог до додатку – сервісів.
 - 3.2. Розробку DTO
 - 3.3. Логіку мапінгу даних на базі бібліотеки Automapper
4. **Розробити функціонал шару WEB/API (контролери).**
Так, зокрема, для відповідного шару тришарової архітектури передбачити:
 - 4.1. Асинхронність, при бажанні, контролерів (<https://docs.microsoft.com/ru-ru/aspnet/core/web-api/action-return-types?view=aspnetcore-5.0>)
 - 4.2. Маршрутизації на основі атрибутів (<https://docs.microsoft.com/ru-ru/aspnet/core/mvc/controllers/routing?view=aspnetcore-5.0>).
 - 4.3. Роботу із статусними кодами;
5. **В якості системи контролю версій використати GIT.**

Контрольні питання:

1. <https://ru.education-wiki.com/8659522-ado.net-interview-questions>
2. <https://www.c-sharpcorner.com/UploadFile/puranindia/ado-net-interview-questions-and-answers/>
3. <https://www.interviewbit.com/ado-net-interview-questions/>