

ПРАКТИЧНЕ ЗАНЯТТЯ № 2

КРИТЕРІЇ ОЦІНКИ ПРОЄКТУ

Вимога	Задов (60+%)	Добре (75+%)	Відмін (90+%)
Організація та запуск			
Проект містить усі артефакти, потрібні для його запуску і перевірки працездатності.	+	+	+
Проект відповідає обраній предметній області та вимогам роботи (наявність відповідних моделей даних).	+	+	+
Розроблено REST Web-сервіс, який коректно запускається.	+	+	+
Внутрішня Архітектура			
Застосунок має внутрішній поділ на три шари: Controller (обробка зовнішніх викликів), Service (бізнес-логіка), Data/Repository .	+	+	+
Міжшарові DTO: У кожному сервісі використовуються DTO для передачі даних між шарами.		+	+
Міжсервісні DTO: При міжсервісній комунікації (OrderService → PaymentService) використовуються спеціалізовані DTO (наприклад, PaymentRequest, PaymentResponse), а не прямі Entity-об'єкти.		+	+
Контракт та документація			
Проект сервісу запускається та надає контракт (WADL, OpenAPI/Swagger).	+	+	+
Розроблено консольний клієнт REST Web-сервісу, який відповідає предметній області.	+	+	+
Проект клієнта запускається та демонструє виклик усіх бізнес-методів Web-сервісу.	+	+	+
Проект клієнта демонструє виклик усіх бізнес-методів Web-сервісу, також з демонстрацією помилок (для перевірки коректності статус кодів відповідей).		±	+
Реалізація REST-методів			
Web-сервіс містить методи, що викликаються по HTTP GET та POST (мінімальний CRUD).	+	+	+
Web-сервіс містить методи, що викликаються по HTTP GET, POST, PUT або PATCH, DELETE (повний CRUD).		±	+

Вимога	Задов (60+%)	Добре (75+%)	Відмін (90+%)
Реалізація ресурсів			
Web-сервіс має методи доступу до ресурсів за ідентифікатором, використовуючи Path Variables (/ {id}) та можливість фільтрації через Query Parameters (?status=...).	±	+	+
Web-сервіс коректно реалізує ієрархію ресурсів (Sub-resource) та/або логіку вкладених ресурсів (наприклад, /orders/ {id}/items).		+	+
Web-сервіс демонструє використання шаблону Resource Locator або еквівалентного механізму для навігації між ресурсами.			+
Інженерні практики			
Реалізовано зберігання даних (persistence) між викликами бізнес-методів (даних предметної області).	− ¹	± ²	+
Усі методи можуть повертати дані у форматах JSON та XML (забезпечено Content Negotiation).	± ³	± ⁴	+
Реалізовано кастомний механізм обробки винятків (Exception Handler) для встановлення коректних статус-кодів відповідей при помилках (наприклад, 404, 400).		+	+
Реалізовано просунуту кастомізацію обробки даних (наприклад, custom message converters) або впровадження контекстних об'єктів (Context Injection).			+

Примітка: Якщо не виконані пункти затінені сірою заливкою, то проєкт взагалі не оцінюється.

¹ Реалізовано як заглушки, що повертають якісь коректні дані, а не null.

² Якщо реалізовано як заглушки, то вони мають повертати якісь коректні дані, а не null.

³ Усі методи можуть повертати дані в одному з форматів JSON **чи** XML.

⁴ Усі методи можуть повертати дані в одному з форматів JSON **чи** XML.