

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «АРХІТЕКТУРА ПЗ»

Лектор курсу			Сватко Віталій Володимирович		Контактна інформація лектора (e-mail), сторінка курсу в Google Classroom		e-mail: v.svatko@duikt.edu.ua сторінка курсу в Google Classroom – https://classroom.google.com/c/ODAYNDM2MjI4NjQ2?cjc=zweavcy6	
Галузь знань			F «Інформаційні технології»		Рівень вищої освіти		бакалавр	
Спеціальність			F2 Інженерія програмного забезпечення		Семестр		5	
Освітня програма			Технології цифрового розвитку		Тип дисципліни		Основна	
3. Обсяг:	Кредитів ECTS	Годин	За видами занять:					
			Лекцій	Семінарських занять	Практичних занять	Лабораторних занять	Самостійна підготовка	
	3	90	18	-	18	-	54	
АНОТАЦІЯ КУРСУ								
Взаємозв'язок у структурно-логічній схемі								
Освітні компоненти, які передують вивченню			Об'єктно-орієнтоване програмування C# Аналіз вимог до програмного забезпечення					
Освітні компоненти для яких є базовою			Моделювання та проектування ПЗ					
Мета курсу:	Мета полягає в формуванні системи понять, знань, умінь і навичок в сучасному програмуванні, зокрема методів аналізу, проектування і створення програмних продуктів з використанням сучасних архітектурних підходів та патернів. Учасники курсу будуть вивчати принципи чистої архітектури, Domain-Driven Design (DDD), Event-Driven Design, SOLID принципи, патерни Repository та Unit of Work, а також сучасні підходи до організації коду та управління залежностями. На завершення курсу студенти будуть здатні проектувати та розробляти масштабовані, підтримувані та тестовані програмні системи з використанням сучасних архітектурних практик на платформі C# і ASP.NET Core, що дозволить їм стати висококваліфікованими програмістами, готовими до роботи з сучасними технологіями та проектами.							

Компетенції відповідно до освітньої програми	
Soft- kills / Загальні компетентності (КЗ)	Hard-skills / Спеціальні компетентності (КС)
.	ПП 2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ПП 3. Здатність розробляти архітектури, модулі та компоненти програмних систем.

Програмні результати навчання (ПР)			
ПРН 10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування. ПРН 11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання. ПРН 12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.			
ОРГАНІЗАЦІЯ НАВЧАННЯ			
Тема, опис теми	Вид заняття	Оцінювання за тему	Форми і методи навчання/питання до самостійної роботи
Розділ 1. Основи архітектури			
Тема 1. Чиста архітектура та Domain-Driven Design Рекомендовані джерела: 1, 2, 3, 4, 7			
Заняття 1.1 Обрати довільну тему яка подобається та створити проєкт ASP.NET Core з використанням чистої архітектури. Визначення bounded contexts і створення моделі взаємодії між доменами. Аналіз архітектурних патернів DDD та їх практичне застосування.	Лекція 1 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 1.2 Створення класів, наповнення їх даними та базою даних, застосування успадкування та модифікаторів доступу. Оголошення конструкторів, віртуальних методів, використання перевизначення та приховання. Інтеграція DDD у чисту архітектуру. Взаємодія шарів Domain і Application. Застосувати моделі які перероблені на DDD та створити класи-сервіси	Практичне заняття 1 2 год		Виконання завдань на практичне застосування знань і вмінь.

Тема 2 Event-Driven Design			
Рекомендовані джерела: 11, 17, 18			
Заняття 2.1. Основи подієво-орієнтованої архітектури. Створення подій і обробників у проєкті ASP.NET Core.	Лекція 2 2 год	10	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 2.2 Event Driven Design. Обрати будь-яку подію в системі (наприклад, створення замовлення, реєстрація користувача, додавання товару) та реалізувати її через Event Driven Design у проєкті ASP.NET Core. Подія має породжувати реакцію в іншому класі/сервісі. Реалізувати у вже наявному проєкті.Створити класи подій (наприклад, Order Created Event.Створити обробки цих подій.	Практичне заняття 2 2 год		Виконання завдань на практичне застосування знань і вмінь.
Тема 1. Чиста архітектура та DDD Тема 2. Event-Driven Design	Самостійна робота		
	12 год		1. Чиста архітектура та Domain-Driven Design (DDD) Побудова шарів застосунку, визначення доменних моделей і бізнес-логіки згідно з принципами DDD. 2. Event-Driven Design Реалізація подій та обробників у застосунку, організація взаємодії між компонентами через події.

Розділ 2. Просунуте програмування			
Тема 3. SOLID принципи <i>Рекомендовані джерела:</i> 1, 4			

Заняття 3.1 Введення в SOLID. Single Responsibility Principle (SRP) - принцип єдиної відповідальності. Приклади порушень та правильної реалізації	Лекція 3 2 год	10	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 3.2 Знайти приклади порушення SOLID у своєму проєкті й виправити. Додати Open-Closed Principle (OCP) - принцип відкритості/закритості з використанням абстракцій та поліморфізму	Практичне заняття 3 2 год		Виконання завдань на практичне застосування знань і вмінь.

Тема 4. Dependency Injection

Рекомендовані джерела: 5, 12

Заняття 4.1. Основи Dependency Injection. Принцип інверсії залежностей і його значення в архітектурі.	Лекція 4 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 4.2. Впровадження DI в ASP.NET Core. Реєстрація сервісів у контейнері. Реалізувати DI контейнер у проєкті <u>ASP.NET</u> Core (zareestruvati servisi z riznymi zhyttsevymy tsyklyamy: Singleton, Scoped, Transient)	Практичне заняття 4 2 год		Виконання завдань на практичне застосування знань і вмінь.

Тема 5. Repository Pattern та Unit of Work

Рекомендовані джерела: 3, 7, 9

Заняття 5.1 Основи Repository Pattern. Абстракція доступу до даних. Інтерфейс IRepository<T>	Лекція 5 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 5.2 Створити для однієї з моделей (наприклад, User або Order): Реалізувати інтерфейс IRepository<T> Додати базові методи: GetById, GetAll, Add, Update, Delete.Створити конкретний репозиторій (наприклад, UserRepository).Продемонструвати використання в контролері. Розширити реалізацію створивши для координації кількох репозиторіїв: Наприклад, IUnitOfWork з методами Users, Orders, SaveChangesAsync().Додати ще один	Практичне заняття 5 2 год		Виконання завдань на практичне застосування знань і вмінь.

репозиторій (наприклад, OrderRepository). Зв'язати обидва репозиторії через Unit of Work. В контролері викликати методи через Unit of Work (наприклад, створити замовлення для користувача)			
---	--	--	--

Розділ 3. DTO

Тема 6. Мапінг об'єктів та розмежування моделей даних

Рекомендовані джерела: 3, 7, 9

Заняття 6.1 Поняття Data Transfer Object (DTO). Призначення DTO для передачі даних між шарами. Відмінності між Domain Models та DTO. Використати AutoMapper у контролерах для конвертації між моделями та DTO при запитах і відповідях. Реалізувати кастомний конвертер у AutoMapper (ITypeConverter).	Лекція 6 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 6.2 Використати DTO для API, підключити AutoMapper. Створити конфігурацію автомаппера для кожної моделі та DTO. Додати складний мапінг між об'єктами з вкладеними властивостями (Nested DTO). Налаштувати двосторонній мапінг (ReverseMap) для обміну даними між domain моделями та DTO. Створити мапінг між кількома рівнями моделей. Налаштувати AutoMapper для роботи зі списками об'єктів (колекціями)	Практичне заняття 6 2 год		Виконання завдань на практичне застосування знань і вмінь.

Розділ 4. Архітектура розподілених систем

Тема 7. Архітектурні стилі та компроміси в розподілених системах

Рекомендовані джерела: 8, 9, 10, 16

Заняття 7.1 Монолітна та модульна архітектура; мікросервісна архітектура; подієво-орієнтовані системи; компроміси між складністю, масштабованістю та підтримуваністю.	Лекція 7 2 год	10	Пояснювально-ілюстративний, лекція-візуалізація.
---	-------------------	----	--

Заняття 7.2 Проаналізувати архітектуру наявного проєкту ASP.NET Core; визначити використаний архітектурний стиль; запропонувати альтернативний архітектурний підхід та обґрунтувати його переваги і недоліки. Спроекувати архітектуру системи, визначивши: архітектурний стиль (моноліт, модульний моноліт або мікросервіси); основні компоненти та їх відповідальності; межі між модулями / сервісами; способи взаємодії між компонентами (REST, події).	Практичне заняття 7 2 год		Виконання завдань на практичне застосування знань і вмінь.
Тема 8. Валідація даних та бізнес-правил <i>Рекомендовані джерела: 6, 7</i>			
Заняття 8.1 Необхідність валідації вхідних даних. Бібліотека FluentValidation. Переваги над Data Annotations	Лекція 8 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 8.2 Додати перевірку даних за допомогою FluentValidation на кожний клас.	Практичне заняття 8 2 год		Виконання завдань на практичне застосування знань і вмінь.
Тема 5. Repository Pattern та Unit of Work Тема 6. DTO та AutoMapper Тема 7. Архітектурні стилі та компроміси в розподілених системах Тема 8. FluentValidation	Самостійна робота		
	24 год		1. LINQ. Фільтрація та вибірка в Repository Pattern Використання LINQ для запитів у репозиторіях, фільтрація й вибірка даних у межах архітектури. 2. Багатопотоковість, синхронізація потоків (TPL) у Unit of Work Паралельне виконання завдань і синхронізація потоків під час роботи кількох репозиторіїв. 3. Ознайомитись з поняттям архітектурних компромісів у розподілених системах (масштабованість, узгодженість, доступність, складність підтримки) та підготувати коротке порівняння вибраних підходів. 4. Самостійно обрати предметну область програмної

			системи та описати можливий архітектурний стиль її реалізації з аргументацією вибору. 5. Підготувати схему взаємодії компонентів або сервісів обраної системи із зазначенням меж відповідальності та способів обміну даними. 6. Ознайомитись з типовими архітектурними помилками в розподілених системах (надмірна складність, передчасна оптимізація, неправильний вибір архітектурного стилю) та сформулювати рекомендації щодо їх уникнення. 7. Асинхронне програмування. Асинхронні операції Реалізація async/await у репозиторіях, сервісах і контролерах ASP.NET Core.
--	--	--	--

Розділ 5 Кешування			
Тема 9. Кешування Рекомендовані джерела: 14, 15			
Заняття 9.1. Основи кешування в архітектурі ПЗ. Типи кешування: In-memory, Distributed.	Лекція 9 2 год	5	Пояснювально-ілюстративний, лекція-візуалізація.
Заняття 9.2 Додати кешування у додаток. Реалізувати кешування за допомогою Memcached	Практичне заняття 9 2 год		Виконання завдань на практичне застосування знань і вмінь.
Тема: DTO та AutoMapper Тема: FluentValidation	Самостійна робота		

Тема Кешування	18 год	1. LINQ. Фільтрація та вибірка 2. DTO та AutoMapper 3. FluentValidation 4. Memcached кешування
----------------	--------	---

МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ	
• Мультимедійний проектор. • Комп'ютерний клас для проведення практичних та лабораторних занять. • Програмне забезпечення: Visual Studio, Visual studio code, платформа .NET (SDK).	
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ	
1. Zhenilevich S. Software Architecture Patterns for Serverless Systems: Architecting for agility and resilience. – Packt Publishing, 2024. 2. Khononov V. Learning Domain-Driven Design: Aligning Software Architecture and Business Strategy. – O'Reilly Media, 2021. 3. Ford N., Parsons R. Fundamentals of Software Architecture: An Engineering Approach. – O'Reilly Media, 2020. 4. Maximiliano C. Clean Code in C#: Refactor your legacy code. – Packt, 2021. 5. Prasad S. Dependency Injection in .NET Core 8.0: Modern Patterns and Practices. – Apress, 2024. 6. Microsoft Docs. ASP.NET Core 8.0 Documentation: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/aspnet/core/?view=aspnetcore-8.0. 7. Microsoft Docs. Cloud-Native .NET Architecture Guide: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/dotnet/architecture/cloud-native/. 8. Microsoft Docs. Architecting Containerized .NET Applications: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/dotnet/architecture/microservices/. 9. Petrov A. Database Internals: A Deep Dive into How Distributed Data Systems Work. – O'Reilly Media, 2023. 10. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd Edition. – O'Reilly Media, 2021. 11. Fowler M. Event-Driven Architecture Patterns: [Електронний ресурс]. – Режим доступу: https://martinfowler.com/articles/202401-event-driven.html 12. Microsoft Docs. Dependency Injection in .NET 8: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/dotnet/core/extensions/dependency-injection. 13. Microsoft Docs. Data Transfer Objects (DTO) in ASP.NET Core: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/aspnet/core/tutorials/first-web-api. 14. Microsoft Docs. Distributed Caching in .NET 8: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/aspnet/core/performance/caching/distributed. 15. Microsoft Docs. Performance and Scalability in Cloud-Native Apps: [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/dotnet/architecture/cloud-native/scale-applications.	

16. Richardson C. Microservices Patterns and Best Practices for Distributed Systems. – Manning Publications, 2023.
17. Microsoft Docs. Serverless Event-Driven Architectures: [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/azure/architecture/guide/architecture-styles/serverless-event-driven>.
18. Hohpe G. The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise. – O'Reilly Media, 2020.

ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)

- Курс передбачає роботу в колективі.
- Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики.
- Освоєння дисципліни передбачає обов'язкове відвідування практичних і лабораторних занять, а також самостійну роботу.
- Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем практичних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблена проробка за рекомендованою літературою.
- Усі завдання, передбачені програмою, мають бути виконані у встановлений термін.
- Якщо студент відсутній з поважної причини, він презентує виконані завдання під час самостійної підготовки та консультації викладача.
- Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання. У разі виявлення факту плагіату студент отримує за завдання 0 балів.
- Студент, який спізнився, вважається таким, що пропустив заняття з неповажної причини з виставленням 0 балів за заняття, і при цьому має право бути присутнім на занятті.
- За використання телефонів і комп'ютерних засобів без дозволу викладача, порушення дисципліни студент видаляється з заняття, за заняття отримує 0 балів.

КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ

Умовою допуску до підсумкового контролю є набрання студентом 30 балів у сукупності за всіма темами дисципліни

Форми контролю	Види навчальної роботи	Оцінювання
ПОТОЧНИЙ КОНТРОЛЬ	Робота на заняттях, у т.ч.:	
	• присутність на заняттях (при пропусках занять з поважних причин допускається відпрацювання пройденого матеріалу)	за кожне відвідування 0,25 бала

	●участь у експрес-опитуванні	за кожну правильну відповідь 0,25 бала
	●виконання завдання (задачі) на практичному занятті	за кожне завдання (задачу) максимум 3 бали
	●виконання завдань (задач) на лабораторному занятті	за кожну лабораторну максимум 3 бали
	●усне опитування, тестування	за кожну правильну відповідь 0,5 бала
	●участь у навчальній дискусії, обговоренні ситуаційного завдання	за кожну правильну відповідь 2 бали
ПІДСУМКОВЕ ОЦІНЮВАННЯ Іспит	Метою іспиту є контроль сформованості практичних навичок та професійних компетентностей, необхідних для виконання професійних обов'язків. Іспит проходить у письмовій формі.	40 балів

ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ

бали	Критерії оцінювання	Рівень компетентності	Оцінка / запис в екзаменаційній відомості
90 - 100	Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях. Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та співставляти дані об'єктів діяльності фахівця на основі набутих з даної та суміжних дисциплін знань та умінь. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань проявив вміння самостійно вирішувати поставлені завдання, активно включатись в дискусії, може відстоювати власну позицію в питаннях та рішеннях, що розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується дисципліни, яка вивчається, але виходить за рамки об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в	Високий Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і вирішенні практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, що відносяться до дисципліни, яка вивчається.	Відмінно / Зараховано (А)

	тлумаченні теоретичних положень чи складних практичних завдань.		
82-89	Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності. Вміє самостійно виправляти допущені помилки, кількість яких є незначною. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.	Достатній Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни	Добре / Зараховано (B)
75-81	Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях/ рішеннях/ розрахунках не є системними. Знає характеристики основних положень, що мають визначальне значення при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, в межах дисципліни, що вивчається.	Достатній Конкретний рівень, за вивченим матеріалом робочої програми дисципліни. Додаткові питання про можливість використання теоретичних положень для практичного використання викликають утруднення.	Добре / Зараховано (C)
67-74	Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядались з викладачем, але допускає значну кількість неточностей і грубих помилок, які може усувати за допомогою викладача.	Середній Забезпечує достатньо надійний рівень відтворення основних положень дисципліни	Задовільно / Зараховано (D)
60-66	Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних/розрахункових завдань дисципліни. Виконання практичних / індивідуальних / контрольних завдань значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами.	Середній Є мінімально допустимим у всіх складових навчальної програми з дисципліни	Задовільно / Зараховано (E)

40-59	Студент може відтворити окремі фрагменти з курсу. Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими. Цілісність розуміння матеріалу з дисципліни у студента відсутні.	Низький Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	Незадовільно з можливістю повторного складання) / Не зараховано (FX) В залікову книжку не проставляється
0-39	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до здачі екзамену/заліку.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни	Незадовільно з обов'язковим повторним вивченням / Не допущений (F) В залікову книжку не проставляється