



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «КОРПОРАТИВНІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Компонента освітньої програми – вибіркова (3 кредити)

Освітньо-професійна програма	Інтелектуальний аналіз даних в комп'ютерних інформаційних системах
Спеціальність	F3 Комп'ютерні науки
Галузь знань	F Інформаційні технології
Рівень вищої освіти	другий (магістр)
Мова навчання	українська
Профайл викладача (-ів)	Томка Юрій Ярославович – кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук https://kkn.chnu.edu.ua/about-kafedra/spivrobitnyky/tomka_yurii/
Контактний тел.	+38095-769-21-91
E-mail:	y.tomka@chnu.edu.ua
Сторінка курсу в Moodle	https://moodle.chnu.edu.ua/course/view.php?id=8317
Консультації	У відповідності до сформованого кафедрою графіків консультацій викладача, що оприлюднений на сторінці: https://kkn.chnu.edu.ua/for_student/class-schedule/

АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Корпоративні архітектури програмного забезпечення» – вибіркова освітня компонента ОП «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах» підготовки здобувачів другого (магістерського) рівня вищої освіти

Метою викладання дисципліни “Корпоративні архітектури програмного забезпечення” є формування у здобувачів системного бачення процесів проєктування, реалізації та еволюції корпоративних програмних систем. Курс забезпечує теоретичну і практичну підготовку з питань стратегічного моделювання підприємства (TOGAF, ArchiMate), вибору та обґрунтування архітектурних стилів і підходів, проєктування дата-центричних та розподілених архітектур, інтеграції систем, забезпечення інфраструктури та управління архітектурною еволюцією.

Особливу увагу приділено практичній роботі з рамковими підходами (TOGAF ADM, ArchiMate 3.2), проєктуванню cloud-native та event-driven систем, побудові архітектур даних (Data Mesh, Data Lake, Lakehouse), інтеграційним патернам та управлінню архітектурними рішеннями через ADR.

Основні завдання курсу

Для досягнення поставленої мети курс передбачає вирішення таких ключових завдань:

- Опанування рамкових підходів до корпоративної архітектури (TOGAF ADM, ArchiMate) для формування узгодженого архітектурного бачення на всіх рівнях підприємства.
 - Вивчення класичних та сучасних архітектурних стилів і підходів: монолітних, мікросервісних, event-driven, serverless та cloud-native систем.
 - Проєктування дата-центричних архітектур (Data Warehouse, Data Lake, Lakehouse, Data Mesh) та забезпечення якості даних через Data Governance.
- Опанування патернів інтеграції та взаємодії систем: API-first підхід, Message-Oriented Middleware, ESB, iPaaS, Saga, Event Streaming.

- Побудова надійної та масштабованої технологічної інфраструктури з використанням контейнеризації, оркестрації, CI/CD та практик Observability.
- Управління архітектурною еволюцією та технічним боргом, прийняття та документування архітектурних рішень (ADR), оцінювання метрик архітектури (DORA).
- Застосування інструментів архітектурного моделювання: Archi, Sparx EA, C4 Model, draw.io.
 - Формування здатності до системного архітектурного мислення та обґрунтування рішень на всіх рівнях підприємства.

Об'єкт вивчення

Об'єктом навчальної дисципліни є процеси та методи проєктування корпоративних програмних систем: стратегічне і тактичне архітектурне планування, вибір та обґрунтування архітектурних стилів і технологій, інтеграція та еволюція систем у контексті бізнес-цілей підприємства.

Практичний підхід та структура навчання

1. Проєктно-орієнтоване навчання

Кожна тема курсу передбачає практичне завдання або лабораторну роботу, де здобувачі моделюють архітектуру корпоративної системи, обирають та обґрунтовують архітектурні стилі, застосовують стандартизовані нотації (ArchiMate, C4 Model, UML) для документування рішень.

2. Наскрізний проєкт

Протягом курсу здобувачі розробляють наскрізний архітектурний проєкт – від бізнес-цілей та архітектурного бачення до вибору архітектурного стилю, проєктування та деталізації окремих рівнів системи (бізнес, дані, застосунки, технології) із застосуванням стандартизованих нотацій (.NET, ArchiMate 3.2, C4-PlantUML, UML, Archi, Enterprise Architect тощо).

НАВЧАЛЬНИЙ КОНТЕНТ ОСВІТНЬОЇ КОМПОНЕНТИ

Змістовий модуль 1. ENTERPRISE ARCHITECTURE: РАМКА ТА МОДЕЛЮВАННЯ	
Тема 1	Вступ до корпоративної архітектури. Рамкові підходи та моделювання ● Поняття корпоративної архітектури (EA): мета, принципи, рівні ● Взаємозв'язок бізнес-стратегії та IT-архітектури ● Огляд рамок підходів: TOGAF, Zachman, FEAF ● TOGAF ADM: фази, артефакти, ролі, Governance ● ArchiMate як мова моделювання корпоративної архітектури ● Рівні ArchiMate: Business, Application, Technology Інструменти: Archi, Enterprise Architect, Modelio, Sparx EA. Практика: побудова базової EA-моделі
Змістовий модуль 2. АРХІТЕКТУРНІ СТИЛІ ТА ПІДХОДИ	
Тема 2	Архітектурний ландшафт та стратегічне планування ● Архітектурний ландшафт: capabilities, value streams, information flows ● Бізнес-архітектура: процеси, організаційна структура, бізнес-функції ● Прикладна архітектура: портфоліо застосунків, інтеграційна карта ● Gap Analysis: поточний стан – цільовий стан – дорожня карта ● Stakeholder Analysis та Viewpoints ● Architecture Roadmap: пріоритети міграції та реалізації Практика: формування архітектурного ландшафту для модельного підприємства
Тема 3	Класичні та розподілені архітектурні стилі ● Монолітна архітектура: переваги, обмеження, сценарії застосування ● Модульний моноліт: межі модулів, ізоляція, публічне API ● Мікросервіси: принципи, гранулярність, комунікація ● SOA та Event-driven архітектура: порівняльний аналіз

	• CQRS та Event Sourcing: розділення відповідальностей читання та запису • Hexagonal Architecture (Ports & Adapters) та Clean Architecture Практика: архітектурне порівняння монолітних та розподілених рішень
Змістовий модуль 3. DATA-CENTRIC АРХІТЕКТУРИ	
Тема 4	Cloud-Native та сучасні архітектурні підходи • Cloud-Native: 12-Factor App та принципи хмарно-орієнтованої розробки • Контейнерна архітектура: Docker, Kubernetes, Helm • Serverless та FaaS: AWS Lambda, Azure Functions, Google Cloud Functions • Service Mesh: Istio, Linkerd – управління мережевою взаємодією • Multi-cloud та Hybrid-cloud: стратегії та виклики • Infrastructure as Code: Terraform, Pulumi, Ansible Практика: проектування cloud-native застосунку на основі 12-Factor
Тема 5	Архітектури зберігання та аналізу даних • Data Warehouse: архітектура, схеми (Star, Snowflake), ETL/ELT-процеси • Data Lake: зберігання сирих даних, каталогізація, зони (Bronze/Silver/Gold) • Lakehouse: поєднання переваг Data Warehouse та Data Lake • Data Mesh: домено-орієнтований підхід до даних як продукту • Streaming архітектури: Apache Kafka, Apache Flink, реальний час • Delta Lake, Apache Iceberg: відкриті формати таблиць Практика: проектування архітектури даних для аналітичної системи
Тема 6	Data Governance та якість даних • Data Governance: поняття, цілі, організаційна структура • Data Catalog: метадані, lineage, класифікація даних • Data Quality: метрики, моніторинг, очищення даних • Регуляторні вимоги: GDPR, Data Residency, PII • Безпека та контроль доступу до даних • Інструменти: Apache Atlas, Collibra, Alation Практика: розробка плану Data Governance для модельної організації
Змістовий модуль 4. ІНТЕГРАЦІЯ ТА ВЗАЄМОДІЯ СИСТЕМ	
Тема 7	API-first підхід та інтеграційні патерни • API-first підхід: проектування та контрактування API • REST, GraphQL, gRPC: порівняльний аналіз та сценарії застосування • OpenAPI Specification (Swagger): документування та тестування API • Інтеграційні патерни: Sync/Async, Request-Reply, Pub/Sub, Consumer • Enterprise Integration Patterns (EIP): агрегатори, маршрутизатори, трансформатори • API Gateway Pattern та API Management • Практика: проектування API-контрактів для розподіленої системи
Змістовий модуль 5. ТЕХНОЛОГІЧНА ІНФРАСТРУКТУРА	
Тема 8	Message-Oriented Middleware та оркестрація транзакцій • Apache Kafka: архітектура, топіки, consumer groups, delivery guarantees • RabbitMQ та Azure Service Bus: порівняльний аналіз • ESB (Enterprise Service Bus): архітектура, переваги та недоліки • iPaaS: MuleSoft, Azure Integration Services, AWS EventBridge • Saga Pattern: оркестрація розподілених транзакцій • Choreography vs Orchestration: вибір підходу • Практика: реалізація асинхронної інтеграції через брокер повідомлень
Змістовий модуль 6. GOVERNANCE ТА ЕВОЛЮЦІЯ АРХІТЕКТУРИ	
Тема 9	Інфраструктурні архітектури та DevOps • Інфраструктурні архітектури: On-premise, Cloud, Hybrid, Multi-cloud • Infrastructure as Code (IaC): Terraform, Pulumi, Ansible • Контейнери та оркестрація: Docker, Kubernetes, Helm • CI/CD Pipeline: GitOps, ArgoCD, GitHub Actions • DevOps та DevSecOps: культура, практики, інструменти • Observability: Logging, Metrics, Tracing – Prometheus, Grafana, OpenTelemetry • Практика: проектування CI/CD pipeline для мікросервісного застосунку
Тема 10	Governance, еволюція архітектури та технічний борг

	• Architecture Governance: принципи, стандарти, процеси, комітети • ADR (Architecture Decision Records): фіксація та обґрунтування архітектурних рішень • Управління змінами: оцінка впливу, версіювання архітектури • Комп'ютерний зір у корпоративних рішеннях • Технічний борг: класифікація, виявлення, пріоритизація, погашення • Метрики архітектури: DORA metrics, якість коду, вартість змін, ризики • Практика: аудит технічного боргу проєкту та план його погашення
--	--

ОСВІТНІ ТЕХНОЛОГІЇ, ФОРМИ ТА МЕТОДИ МЕТОДИ НАВЧАННЯ

У процесі вивчення навчальної дисципліни використовуються інноваційні освітні технології: інформаційно-комунікаційні, технології студентоцентрованого навчання; проєктна діяльність; традиційні та інтерактивні форми і методи навчання, серед яких: лекція-візуалізація, проблемна лекція, семінар-дискусія, семінар-діалог, самостійно-дослідницька робота, лабораторна робота, аналіз і рішення ситуативних професійних психолого-педагогічних задач (Case study) та ін.

ФОРМИ Й МЕТОДИ КОНТРОЛЮ ТА ОЦІНЮВАННЯ

Поточний контроль: усні та письмові (тестування) відповіді здобувача; представлення презентації, доповідь та аргументований захист у процесі командної роботи; захист лабораторних робіт та наскрізного архітектурного проєкту.

Підсумковий контроль – залік.

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання програмних результатів навчання здобувачів освіти здійснюється за шкалою європейської кредитно-трансферної системи (ECTS).

Критерієм успішного оцінювання є досягнення здобувачем вищої освіти мінімальних порогових рівнів (балів) за кожним запланованим результатом навчання.

ПОЛІТИКА ЩОДО АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

- «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» <https://www.chnu.edu.ua/media/jxpbs0zb/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu.pdf> ;

- «Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича» https://www.chnu.edu.ua/media/f5eleobm/polozhennya-pro-zapobihannia-plahiatu_2024.pdf.

ІНФОРМАЦІЙНІ РЕСУРСИ

Основна література

1. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. — O'Reilly Media, 2020. — 419 p.
<https://dataintensive.net/>
2. Kleppmann M. Designing Data-Intensive Applications. — O'Reilly Media, 2017. — 611 p.

- <https://dataintensive.net/>
3. Richards M., Ford N. Software Architecture: The Hard Parts. — O'Reilly Media, 2022. — 459 p.
<https://www.oreilly.com/library/view/software-architecture-the/9781492086888/>
 4. The Open Group. TOGAF® Standard, 10th Edition. — The Open Group, 2022.
<https://www.opengroup.org/togaf>
 5. Kleppmann M. Designing Data-Intensive Applications. — O'Reilly Media, 2017. — 611 p.
<https://dataintensive.net/>
 6. The Open Group. TOGAF® Standard, 10th Edition. — The Open Group, 2022.
<https://www.opengroup.org/togaf>
- Додаткова література**
7. Bellemare A. Building Event-Driven Microservices. — O'Reilly Media, 2020. — 322 p.
<https://www.oreilly.com/library/view/building-event-driven-microservices/9781492057888/>
 8. Indrasiri K., Suhothayan S. Design Patterns for Cloud Native Applications. — O'Reilly Media, 2021. — 334 p.
<https://www.oreilly.com/library/view/design-patterns-for/9781492090717/>
 9. Khononov V. Learning Domain-Driven Design. — O'Reilly Media, 2021. — 340 p.
<https://www.oreilly.com/library/view/learning-domain-driven-design/9781098100124/>
 10. Kim G., Humble J. та ін. The DevOps Handbook (2nd ed.). — IT Revolution Press, 2021. — 528 p.
<https://itrevolution.com/product/the-devops-handbook-second-edition/>

Електронні інформаційні ресурси

1. Domain-Driven Design Reference [Електронний ресурс] // Eric Evans. – 2015. – Режим доступу: <https://www.opengroup.org/togaf>
2. eShopOnContainers – DDD Reference Application [Електронний ресурс] // Microsoft. – 2023. – Режим доступу: <https://www.opengroup.org/archimate-forum/archimate-overview>
3. Martin Fowler's DDD Blog [Електронний ресурс] // Martin Fowler. – 2024. – Режим доступу: <https://aws.amazon.com/architecture/well-architected/>
4. Modular Monolith: A Primer [Електронний ресурс] // Kamil Grzybek. – 2020. – Режим доступу: <https://docs.microsoft.com/azure/architecture/>
5. MediatR Documentation [Електронний ресурс] // Jimmy Bogard. – 2024. – Режим доступу: <https://landscape.cncf.io/>
6. Microsoft .NET Architecture Guides [Електронний ресурс] // Microsoft. – 2024. – Режим доступу: <https://martinfowler.com/architecture/>
7. EventStoreDB Documentation [Електронний ресурс] // Event Store Ltd. – 2024. – Режим доступу:
<https://adr.github.io/>
8. CQRS Journey – Microsoft patterns & practices [Електронний ресурс] // Microsoft. – 2012. – Режим доступу:
<https://dora.dev/>
9. Awesome Domain-Driven Design [Електронний ресурс] // GitHub community. – 2024. – Режим доступу: <https://12factor.net/>
10. Entity Framework Core Documentation [Електронний ресурс] // Microsoft. – 2024. – Режим доступу: <https://dora.dev/>

Детальна інформація щодо вивчення курсу «Корпоративні архітектури програмного забезпечення» висвітлена у робочій програмі навчальної дисципліни, що наведена на сторінці:

https://kkn.chnu.edu.ua/for_student/course-programs/

