



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«ПРЕДМЕТНО-ОРІЄНТОВАНА РОЗРОБКА ПРОГРАМНИХ СИСТЕМ (DOMAIN-DRIVEN DESIGN)»

Компонента освітньої програми – вибіркова (3 кредити)

Освітньо-професійна програма	Інтелектуальний аналіз даних в комп'ютерних інформаційних системах
Спеціальність	122 Комп'ютерні науки
Галузь знань	12 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 «Предметно-орієнтована розробка програмних систем (Domain-Driven Design)» – вибіркова освітня компонента ОП «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах» підготовки здобувачів другого (магістерського) рівня вищої освіти

Метою викладання дисципліни "Предметно-орієнтована розробка програмних систем (Domain-Driven Design)" є формування у студентів системного підходу до проєктування складного програмного забезпечення через призму предметної моделі. Курс надає теоретичні та практичні навички стратегічного аналізу, моделювання предметної області, а також реалізації програмних систем із застосуванням паттернів тактичного DDD – Агрегатів, Сутностей, Об'єктів-значень, Доменних подій та Репозиторіїв.

Особливу увагу приділено побудові модульного моноліту з використанням принципів чистої архітектури (Clean Architecture), організації міжмодульної взаємодії через внутрішні повідомлення та контракти, а також практичним аспектам стратегічного картування (Event Storming, Context Mapping).

Основні завдання курсу

Для досягнення поставленої мети курс передбачає вирішення таких ключових завдань:

- Опанування стратегічних концепцій DDD:
 - Вивчення Ubiquitous Language, Bounded Context та Context Map.
 - Event Storming як метод дослідження предметної області.
- Реалізація тактичних паттернів DDD:
 - Проєктування Агрегатів, Сутностей та Об'єктів-значень.
 - Доменні події, Репозиторії та Доменні сервіси у практичних завданнях.
- Побудова модульного моноліту на основі чистої архітектури:
 - Організація шарів: Domain, Application, Infrastructure, Presentation.
 - Міжмодульна комунікація: внутрішні події, інтеграційні контракти, публічне API модулів.

- Практичне застосування CQRS та Event Sourcing:
 - Розділення відповідальностей читання та запису (CQRS).
 - Зберігання стану системи через потік подій (Event Sourcing).

Об'єкт вивчення

Об'єктом навчальної дисципліни є процеси та методи проєктування складного програмного забезпечення, зокрема стратегічні та тактичні практики Domain-Driven Design, архітектурні рішення у побудові модульних систем, а також способи організації коду та взаємодії між компонентами системи.

Практичний підхід та структура навчання

1. Проєктно-орієнтоване навчання

Кожна тема курсу передбачає практичне завдання або лабораторну роботу, де студенти моделюють предметну область реального застосунку, проєктують систему за принципами DDD та реалізують відповідні компоненти з використанням обраного технологічного стеку.

2. Наскрізний проєкт

Протягом курсу студенти розробляють наскрізний проєкт – модульний моноліт для конкретної предметної області (наприклад, система замовлень, система управління проєктами), який охоплює стратегічне картування, реалізацію доменної моделі та організацію міжмодульної комунікації з використанням (.NET, Java, Python, Node.js, Golang тощо).

НАВЧАЛЬНИЙ КОНТЕНТ ОСВІТНЬОЇ КОМПОНЕНТИ

Змістовий модуль 1. Стратегічний аналіз та основи Domain-Driven Design	
Тема 1	Вступ до Domain-Driven Design. Стратегічний аналіз предметної області • Передумови виникнення DDD: складність та ентропія у програмних системах • Ключові принципи DDD: модель як центр системи • Стратегічний рівень DDD: огляд концепцій • Ubiquitous Language: спільна мова команди та бізнесу • Bounded Context: визначення меж предметної моделі • Domain, Subdomain, Core Domain: класифікація підобластей Практична вправа: виявлення підобластей у реальному бізнес-сценарії
Тема 2	Event Storming як метод дослідження предметної області • Принцип “Big Picture” Event Storming: виявлення подій системи • Деталізований Event Storming: команди, актори, агрегати • Виявлення Bounded Contexts у процесі сесії • Хот-споти та нерозв’язані питання предметної області • Process Modelling: моделювання бізнес-процесів • Практична сесія Event Storming на навчальному прикладі Документування результатів сесії
Тема 3	Bounded Context та Context Mapping: взаємодія між контекстами • Context Map: типи взаємозв’язків між контекстами • Паттерни Context Mapping: Shared Kernel, Customer-Supplier, Conformist • Anti-Corruption Layer: захист від забруднення моделі • Open Host Service та Published Language • Практика: побудова Context Map для реального проєкту • Вибір стратегії інтеграції між контекстами Еволюція та рефакторинг Context Map
Тема 4	Тактичні паттерни DDD: Сутності та Об’єкти-значень • Сутності (Entities): ідентичність та рівність • Об’єкти-значень (Value Objects): незмінюваність та виразність • Проєктування Value Objects у мові C#/.NET • Порівняння Entities та Value Objects: коли що обирати • Валідація та інваріанти у тактичних паттернах

	Практика: моделювання Сутностей та Value Objects Code review: типові помилки при проектуванні
Тема 5	Агрегати: проектування меж транзакційної узгодженості - Aggregate Root: правила доступу та захист інваріантів - Правила розміру Агрегату: один Агрегат – одна транзакція - Посилання між Агрегатами через ідентифікатори - Завантаження та збереження Агрегатів через Репозиторій - Практика: рефакторинг Агрегатів на прикладі - Доменні Сервіси (Domain Services): відповідальність без стану Специфікації (Specifications) та їх застосування
Тема 6	Репозиторії та Доменні події - Repository Pattern: абстракція доступу до даних - Реалізація Репозиторіїв у .NET з EF Core - Доменні події (Domain Events): моделювання фактів - Публікація та обробка Доменних подій - Eventual Consistency через доменні події - Integration Events: відмінність від Domain Events Практика: реалізація Репозиторіїв та Доменних подій
Архітектура та реалізація програмних систем на основі DDD	
Тема 7	Чиста архітектура (Clean Architecture) у поєднанні з DDD - Концепція чистої архітектури: шари та залежності - Domain Layer: незалежність від інфраструктури - Application Layer: Use Cases та Application Services - Infrastructure Layer: реалізація репозиторіїв, ORM, зовнішні сервіси - Presentation Layer: контролери, DTO, мапінг - Dependency Rule: правило напряму залежностей - Практика: структура проекту Clean Architecture + DDD
Тема 8	Модульний моноліт: структура, межі та організація модулів - Поняття модульного моноліту та переваги перед мікросервісами - Визначення меж модулів на основі Bounded Contexts - Структура модуля: публічне API та внутрішня реалізація - Управління залежностями між модулями - Фізична та логічна ізоляція модулів у .NET - Спільна інфраструктура: єдина БД vs окремі схеми - Практика: проектування структури модульного моноліту
Тема 9	Міжмодульна комунікація: внутрішні події та контракти - Синхронна комунікація між модулями через публічне API - Асинхронна комунікація через in-process повідомлення (MediatR) - Integration Events: контракти між модулями - Outbox Pattern: надійна публікація подій - Уникання циклічних залежностей між модулями - Тестування міжмодульної взаємодії - Практика: реалізація міжмодульної комунікації через MediatR
Тема 10	CQRS та Event Sourcing - Command Query Responsibility Segregation: розділення моделей - Read Model та Write Model: окремі стеки та оптимізація - Event Sourcing: збереження стану як потік подій - Комп'ютерний зір у корпоративних рішеннях - Проекції (Projections): побудова Read Model з подій - Snapshots: оптимізація Event Sourcing для великих Агрегатів - Практика: реалізація CQRS + Event Sourcing у .NET

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 студента; представлення презентації, доповідь та аргументований захист у процесі командної роботи.

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/jxdfs0zb/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. Millett S., Tune N. Patterns, Principles, and Practices of Domain-Driven Design. — Wrox, 2015. — 792 p.
<https://www.amazon.com/Patterns-Principles-Practices-Domain-Driven-Design/dp/1118714709>
2. Millett S., Tune N. Patterns, Principles, and Practices of Domain-Driven Design. — Wrox, 2015. — 792 p.
<https://www.amazon.com/Patterns-Principles-Practices-Domain-Driven-Design/dp/1118714709>
3. Vernon V. Vaughn Vernon on Domain-Driven Design. — Addison-Wesley, 2016. — 374 p.
<https://www.amazon.com/Vaughn-Vernon-Domain-Driven-Design-Distilled/dp/0134434420>
4. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 560 p.
<https://www.amazon.com/Patterns-Enterprise-Application-Architecture-Martin/dp/0321127420>
5. Millett S., Tune N. Patterns, Principles, and Practices of Domain-Driven Design. — Wrox, 2015. — 792 p.

<https://www.amazon.com/Patterns-Principles-Practices-Domain-Driven-Design/dp/1118714709>

6. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 560 p.
<https://www.amazon.com/Patterns-Enterprise-Application-Architecture-Martin/dp/0321127420>

Додаткова література

7. Richardson C. Microservices Patterns: With examples in Java. — Manning Publications, 2018. — 520 p.
<https://www.amazon.com/Microservices-Patterns-examples-Chris-Richardson/dp/1617294543>
8. Wlaschin S. Domain Modeling Made Functional. — Pragmatic Bookshelf, 2018. — 320 p.
<https://www.amazon.com/Domain-Modeling-Made-Functional-Domain-Driven/dp/1680502549>
9. Young G. CQRS Documents. — Leanpub, 2010. — 110 p.
https://cqrs.files.wordpress.com/2010/11/cqrs_documents.pdf
10. Brandolini A. Introducing EventStorming. — Leanpub, 2021. — 380 p.
https://leanpub.com/introducing_eventstorming

Електронні інформаційні ресурси

1. Domain-Driven Design Reference [Електронний ресурс] // Eric Evans. – 2015. – Режим доступу: https://domainlanguage.com/wp-content/uploads/2016/05/DDD_Reference_2015-03.pdf
2. eShopOnContainers – DDD Reference Application [Електронний ресурс] // Microsoft. – 2023. – Режим доступу: <https://github.com/dotnet-architecture/eShopOnContainers>
3. Martin Fowler's DDD Blog [Електронний ресурс] // Martin Fowler. – 2024. – Режим доступу: <https://martinfowler.com/tags/domain%20driven%20design.html>
4. Modular Monolith: A Primer [Електронний ресурс] // Kamil Grzybek. – 2020. – Режим доступу: <https://www.kamilgrzybek.com/design/modular-monolith-primer/>
5. MediatR Documentation [Електронний ресурс] // Jimmy Bogard. – 2024. – Режим доступу: <https://github.com/jbogard/MediatR>
6. Microsoft .NET Architecture Guides [Електронний ресурс] // Microsoft. – 2024. – Режим доступу: <https://dotnet.microsoft.com/en-us/learn/dotnet/architecture-guides>
7. EventStoreDB Documentation [Електронний ресурс] // Event Store Ltd. – 2024. – Режим доступу:
<https://developers.eventstore.com/>
8. CQRS Journey – Microsoft patterns & practices [Електронний ресурс] // Microsoft. – 2012. – Режим доступу:
[https://learn.microsoft.com/en-us/previous-versions/msp-n-p/jj554200\(v=pandp.10\)](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/jj554200(v=pandp.10))
9. Awesome Domain-Driven Design [Електронний ресурс] // GitHub community. – 2024. – Режим доступу: <https://github.com/heynickc/awesome-ddd>
10. Entity Framework Core Documentation [Електронний ресурс] // Microsoft. – 2024. – Режим доступу: <https://learn.microsoft.com/en-us/ef/core/>

Детальна інформація щодо вивчення курсу «Назва навчальної дисципліни» висвітлена у робочій програмі навчальної дисципліни, що наведена на сторінці:

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