

СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	SE008 Архітектура програмного забезпечення / Software Architecture
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
Освітня програма	Інженерія програмного забезпечення
Семестр	3 семестр (на базі повної загальної середньої освіти) 5 семестр (на базі базової середньої освіти)
Курс	2 курс (на базі повної загальної середньої освіти) 3 курс (на базі базової середньої освіти)
Анотація курсу	Цей курс знайомить студентів з поняттями та принципами, які стосуються архітектури та проектування програмного забезпечення, формує базові навички з вироблення архітектурних артефактів у ході життєвого циклу розробки програмного забезпечення. Курс охоплює питання ролі архітекторів у програмних проєктах та передумов для виконання їх основні обов'язків; прийняття та документування ключових архітектурних рішень, що визначають ключові технології та атрибути якості до програмного проєкту, що розробляється. Практична частина курсу пов'язана з технологічним стеком .NET та реалізацією на його рівні архітектурних принципів та шаблонів.
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=219
Мова викладання	українська
Лектор курсу	Марченко Станіслав Віталійович, викладач, спеціаліст першої категорії канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: sv.marchenko1989@gmail.com
Місце дисципліни в освітній програмі	
Освітня програма	http://csbc.edu.ua/documents/otdel/moop_pr24.pdf
Інтегральна компетентність	Здатність вирішувати типові спеціалізовані задачі інженерії програмного забезпечення, що вимагає застосування положень і методів відповідних наук

Базова інформація про дисципліну	
Назва дисципліни	SE008 Архітектура програмного забезпечення / Software Architecture
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
	(математики, інформатики, інформаційних технологій, тощо) та може характеризуватися певною невизначеністю умов; нести відповідальність за результати своєї діяльності; здійснювати контроль інших осіб у визначених ситуаціях.
Перелік загальних компетентностей (ЗК)	ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності. ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. ЗК07. Здатність застосовувати знання у практичних ситуаціях.
Перелік спеціальних компетентностей (СК)	СК01. Здатність алгоритмічно та логічно мислити. СК02. Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя. СК04. Здатність дотримуватися стандартів при розробці програмного забезпечення. СК06. Здатність брати участь у проектуванні програмного забезпечення. СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів.
Перелік програмних результатів навчання	РН02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення. РН03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення. РН05. Розробляти та супроводжувати програмне забезпечення. РН07. Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення. РН08. Аналізувати вимоги до програмного забезпечення. РН09. Розуміти основні принципи командної роботи при розробці програмного забезпечення. РН10. Обирати та застосовувати ефективні методи оптимізації алгоритмів. РН11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем. РН14. Розуміти предметну область, застосовувати знання у професійній діяльності. РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання

Базова інформація про дисципліну	
Назва дисципліни	SE008 Архітектура програмного забезпечення / Software Architecture
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
	професійних задач з урахуванням сучасних досягнень інформаційних технологій.
Опис дисципліни	
Структура навантаження на студента	Загальна кількість годин – 90 Кількість кредитів – 3 Кількість лекційних годин – 30 Кількість практичних занять – 30 Кількість годин для самостійної роботи студентів – 30 Форма підсумкового контролю – іспит
Методи навчання	За подачею навчального матеріалу: методи готових знань, дослідницький метод. З огляду на мету навчання: методи здобуття нових знань, метод формування умінь і навичок, метод застосування знань на практиці, методи закріплення знань, умінь і навичок, методи перевірки і оцінювання знань, умінь і навичок.
Зміст дисципліни	
Тема 1. Архітектура та проектування ПЗ. Узагальнення базових концепцій ООП.	Базові концепції архітектури та проектування програмного забезпечення. Ознаки якісного API. Поняття зв'язаності (coupling) та згуртованості (cohesion). SOLID-принципи розробки об'єктно-орієнтованого коду. Побудова наративної об'єктної моделі. Побудова UML-діаграм засобами PlantUML.
Тема 2. Вступ до патернів проектування	Патерни проектування: поняття, опис та класифікація. Проблеми, які вирішують патерни проектування. Вибір та використання патерну. Твірні та структурні патерни проектування. Шаблон «Будівельник». Фабричні шаблони: «фабричний метод» та «Абстрактна фабрика» Шаблон «Прототип». Шаблон «Одиночка». Шаблон «Адаптер». Шаблон «Міст». Шаблон «Компонувальник». Шаблон «Декоратор». Шаблон «Фасад». Шаблон «Легковаговик». Шаблон «Заступник». Поведінкові патерни проектування. Шаблон «Команда». Шаблон «Ланцюжок обов'язків». Шаблон «Ітератор». Шаблон «Посередник». Шаблон «Знімок». Шаблон «Спостерігач». Шаблон «Стан». Шаблон «Стратегія». Шаблон «Шаблонний метод». Шаблон «Відвідувач».

Базова інформація про дисципліну	
Назва дисципліни	SE008 Архітектура програмного забезпечення / Software Architecture
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
Тема 3. Основні види діяльності та документації в архітектурі та проектуванні програмного забезпечення	Основні види діяльності в галузі архітектури програмного забезпечення. Архітектурні структури та стилі. Характеристики та архітектурні принципи сучасних вебдодатків. Загальні архітектурні стилі вебдодатків. Візуалізація архітектури за допомогою моделі C4 та PlantUML. Методика QAW та ATAM-аналіз. Solution Architecture Document.
Тема 4. Принципи та патерни керування залежностями	Принцип інверсії управління та його прояви. Шаблон «Впровадження залежностей». Поняття та принципи реалізації. DI-патерни. DI-антипатерни та запахи коду. Використання шаблону «Впровадження залежностей» в ASP.NET Core. Модульна контрольна робота 1.
Тема 5. Монолітна архітектура на базі ASP.NET Core MVC	Загальні відомості про конкурентне виконання коду. Асинхронне виконання коду. Шаблон MVC. Класи-контролери. Прив'язування моделі. Робота з рівнем представлення. Дані як аспект архітектури. Робота з даними в ASP.NET Core додатках.
Тема 6. Атрибути якості та використання проміжних компонентів ASP.NET Core	Безпека як аспект архітектури. Масштабованість як аспект архітектури. Продуктивність як аспект архітектури. Стійкість (resilience) як аспект архітектури.
Політика дисципліни	
Політика відвідування	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в он-лайн формі за погодженням із керівником курсу.
Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.
Використання ІІІ	Використання ІІІ під час виконання завдань регламентується Положенням «Про використання ІІІ в освітньому процесі ЧДБК». Завдання містять маркування щодо регламенту використання ІІІ.
Підсумковий контроль	Іспит

Система оцінювання

Система оцінювання підсумкової успішності студентів поділяється на **поточний контроль** та **семестровий контроль**.

Поточний контроль здійснюється протягом семестру і охоплює всі види аудиторної роботи (практичні, семінарські заняття) та виконання індивідуальних завдань. Максимальна кількість балів, яку студент може набрати за цей вид контролю, становить 100.

Підсумковий контроль

Відбувається в формі іспиту.

Розрахунок підсумкової оцінки

Підсумкова оцінка (O) розраховується як сума балів за роботу в семестрі (S) та балів за іспит (T), кожен з яких має максимальну вагу 100 балів. Вагові коефіцієнти для обох компонентів однакові та дорівнюють **0,5**.

Формула: $O = (S \times 0.5) + (T \times 0.5)$

Види навчальної роботи	Загальна кількість балів
Усні відповіді за темою 2 по 5 балів	10
Практична робота за темами 1, 3, 6 по 10 балів	30
за темою 4 по 6 балів	6
за темою 5 по 12 балів	12
Модульні контрольні (2 к.р. по 6 балів)	12
Індивідуальна самостійна робота (проєкт)	30
Разом	100

Критерії оцінювання для кожного виду навчальної

Критерії оцінювання усних відповідей:

3 б. Підготовані презентаційні матеріали містять докладний опис шаблону проєктування, ситуації застосування, робочий програмний код з реалізацією та демонстрацією шаблону; студент послідовно та логічно висвітлює поставлене завдання, правильно відповідає на запитання, здатний до критичного аналізу та порівняння шаблонів.

2 б. Підготовані презентаційні матеріали містять докладний опис шаблону проєктування, ситуації застосування; студент послідовно та логічно висвітлює поставлене завдання, проте частково правильно відповідає на запитання, має обмежене розуміння контексту використання шаблонів.

1 б. Підготовані презентаційні матеріали містять докладний опис шаблону проєктування; студент має непослідовності в викладі інформації, частково правильно відповідає на запитання, не має розуміння контексту використання шаблонів.

Критерії оцінювання практичних робіт

100% – звіт з практичної роботи подано вчасно з дотриманням умов академічної доброчесності, термінів звітування, вимог до оформлення та коректності документування, всі завдання розв'язано, захищено усно (якщо передбачено умовою завдань).

90-99% – звіт з практичної роботи подано вчасно з дотриманням умов академічної доброчесності, термінів звітування, наявністю зауважень до оформлення та коректності документування, всі завдання розв'язано, захищено усно (якщо передбачено умовою завдань).

80-89% – звіт з практичної роботи подано вчасно з дотриманням умов академічної доброчесності, термінів звітування, наявністю значних зауважень до оформлення та коректності документування, всі завдання розв'язано, захищено усно (якщо передбачено умовою завдань).

50-79% – звіт з практичної роботи подано вчасно з дотриманням умов академічної доброчесності, термінів звітування, вимог до оформлення та коректності документування, всі завдання розв'язано, проте не захищено усно (якщо це передбачено умовою завдань).

Відсотковий показник визначається стосовно сумарної кількості балів, отриманої за практичну роботу за результатами звітування. Порушення умов академічної доброчесності та/або визначених термінів звітування може позбавити здобувача освіти права подачі звіту за відповідною темою.

Критерії оцінювання модульних робіт

Модульні контрольні роботи складаються з набору закритих тестових завдань та автоматично перевіряються засобами СДН Moodle.

Критерії оцінювання індивідуальних робіт (проектів)

Індивідуальні роботи охоплюють питання доменного аналізу, розбудови архітектури та імплементації програмного рішення для вебплатформи з заданої предметної області.

30 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) з дотриманням умов академічної доброчесності та термінів звітування проаналізував вимоги до програмного забезпечення, побудував репозиторій архітектурних рішень на основі методики QAW та АТАМ-аналізу; розробив набір UML/C4-діаграм для архітектурного опису структури та поведінки системи; програмно реалізував відповідний вебсайт засобами фреймворку ASP.NET Core та виконав умови до демонстрації та звітування за проектом.

20-29 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) з дотриманням умов академічної доброчесності та термінів звітування проаналізував вимоги до програмного забезпечення, побудував репозиторій архітектурних рішень на основі методики QAW та АТАМ-аналізу; розробив набір UML/C4-діаграм для архітектурного опису структури та поведінки системи; проте лише частково програмно реалізував відповідний вебсайт засобами фреймворку ASP.NET Core та виконав умови до демонстрації та звітування за проектом.

15-19 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) з дотриманням умов академічної доброчесності та термінів звітування проаналізував вимоги до програмного забезпечення, побудував репозиторій архітектурних рішень на основі методики QAW та АТАМ-аналізу; розробив набір UML/C4-діаграм для архітектурного опису структури та поведінки системи та виконав умови до демонстрації та звітування за проектом.

10-14 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) з дотриманням умов академічної доброчесності та термінів звітування проаналізував вимоги до програмного забезпечення, побудував репозиторій архітектурних рішень на основі методики QAW та АТАМ-аналізу та виконав умови до демонстрації та звітування за проектом.

1-9 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) з дотриманням умов академічної доброчесності та термінів звітування проаналізував вимоги до програмного забезпечення та виконав умови до демонстрації та звітування за проектом.

0 б. Здобувач освіти особисто чи в складі мінігрупи (до 4 осіб) не дотримався умов академічної доброчесності та термінів звітування або не виконав поставлені завдання.

Критерії оцінювання іспиту

За екзамен з предмету можна набрати 100 балів максимально. Білет складається з 3 питань, одного теоретичного та двох теоретико-практичних. За відповідь на теоретичне питання можна отримати максимально 30 балів, виконане теоретико-практичне завдання – 35 балів.

Таблиця 1. Бали за відповідь на теоретичне питання

Бали	Критерії
30	Студент (студентка) правильно, повно, чітко і логічно відповідає на всі поставлені питання
20-29	Правильно і повно, інколи з деякою неточністю та з допомогою навідних (пояснювальних) питань відповідає на всі поставлені питання
1-19	Відповідь студента (студентки) неповна і неточна, на навідні питання відповідає не зовсім точно, відповідь не є чіткою і логічною
0	Не може відповісти на питання взагалі

Таблиця 2. Бали за виконання теоретико-практичного завдання

Бали	Критерії
35	Студент (студентка) самостійно деталізує вхідні дані до завдання, аналізує алгоритм виконання завдання та особливості реалізації. Програмна реалізація чи візуалізація повністю відповідає поставленому завданню, і студент (студентка) здатний вносити в неї невеликі корективи.
21-34	Студент (студентка) самостійно деталізує вхідні дані до завдання, але не може обґрунтувати їх доцільність, аналізує алгоритм виконання завдання. Програмна реалізація чи візуалізація переважно відповідає поставленому завданню.

6-20	Студент (студентка) знає алгоритм виконання, потрібні синтаксичні інструменти. Програмна реалізація чи візуалізація частково відповідає поставленому завданню.
1-5	Студент (студентка) знає алгоритм виконання, потрібні синтаксичні інструменти, проте програмну реалізацію чи візуалізацію здійснити не може.
0	Завдання не виконано взагалі

Шкала оцінювання		
ECTS	Бали	Зміст
A	90-100	Бездоганна підготовка в широкому контексті
B	80-89	Повні знання, міцні вміння
C	70-79	Хороші знання та вміння
D	65-69	Задовільні знання, стереотипні вміння
E	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
FX	35-59	Слабкі знання, відсутність умінь
F	1-34	Необхідний повторний курс

Список рекомендованих джерел

1. Alls J. Clean Code in C#. Refactor your legacy C# code base and improve application performance by applying best practices. Birmingham: Packt Publishing, 2020. 472p.
2. Sazanavets F. The easiest way to learn design patterns With C# code samples using .NET 6 templates. Lean Publishing, 2022. 326c.
3. Ford N. et al. Building Evolutionary Architectures: Automated Software Governance, 2nd Edition. Sebastopol: O'Reilly Media, 2022. 262p. ISBN: 9781492097549.
4. Erder M., Pureur P., Woods E. Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps. Boston: Addison-Wesley, 2021. 322c.
5. Bass L., Clements P., Kazman R. Software Architecture in Practice (SEI Series in Software Engineering), 4th Edition. Boston: Addison-Wesley, 2021. 442p.
6. Richards M., Ford N. Fundamentals of Software Architecture: A Modern Engineering Approach, Second Edition. Sebastopol: O'Reilly Media, 2025. 482p.
7. Jager E. Getting Started with Enterprise Architecture: A Practical and Pragmatic Approach to Learning the Basics of Enterprise Architecture. NY: Appress, 2023. 276p.
8. Harmet-Law A. Facilitating Software Architecture: Empowering Teams to Make Architectural Decisions. Sebastopol: O'Reilly Media, 2025. 482p.
9. Smith S. Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure. Redmond, Washington: Microsoft Developer Division, .NET, and Visual Studio product teams, 2022.
10. Richards M. Software Architecture Patterns. Understanding Common Architectural Styles and When to Use Them. Sebastopol: O'Reilly Media, 2022. 68p.
11. Vázquez-Ingelmo A., García-Holgado, A., García-Peñalvo, F. J. C4 model in a Software Engineering subject to ease the comprehension of UML and the software development process. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 27-30 April 2020, Porto, Portugal. pp. 919-924. DOI: 10.1109/EDUCON45650.2020.9125335.
12. Baptista G., Abbruzzese F. Software Architecture with C# 12 and .NET 8. Build enterprise applications using microservices, DevOps, EF Core, and design patterns for Azure, Fourth Edition. Birmingham: Packt Publishing, 2022. 714p.

Допоміжна література

1. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення. Харків: Фабула, 2019. 416с.
2. Bell M. Software Architect. Hoboken: Wiley, 2023. 388p.
3. Nesteruk D. Design Patterns in .NET. Reusable Approaches in C# and F# for Object-Oriented Software Design: How to Build Better Software Faster. NY: Appress, 2019. 360с.
4. van Deursen S., Seemann S. Dependency Injection: Principles, Practices, and Patterns. NY, Shelter Island: Manning Publications, 2019. 532с.
5. Sarcar V. Design Patterns in C#: A Hands-on Guide with Real-World Examples. NY: Appress, 2018. 456p.
6. Kumar R., Khan S. A., Khan R. A. Software Durability: Concepts and Practices. Boca Raton: CRC Press, 2023. 330p.
7. Garverick J., McIver O. D. Implementing Event-Driven Microservices Architecture in .NET 7. Develop event-based distributed apps that can scale with ever-changing business demands using C# 11 and .NET 7. Birmingham: Packt Publishing, 2023. 306p.

Інтернет ресурси

1. Каталог патернів проектування [Електронний ресурс]. URL: <https://refactoring.guru/uk/design-patterns/catalog>.
2. Catalog of Patterns of Enterprise Application Architecture [Електронний ресурс]. URL: <https://martinfowler.com/eaCatalog/>.
3. Design Patterns in C# With Real-Time Example [Електронний ресурс]. URL: <https://dotnettutorials.net/course/dot-net-design-patterns/>.
4. Architectural Katas [Електронний ресурс]. URL: <https://www.architecturalkatas.com/#>.