



СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	SE008 / Архітектура програмного забезпечення / Software Architecture
Рівень вищої освіти / фахової передвищої освіти	Фахова передвища
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
Освітня програма	Інженерія програмного забезпечення
Семестр	7 семестр (9 кл), 5 семестр (11 кл)
Факультет / відділення	Інженерії програмного забезпечення
Курс	4 курс (9 кл), 3 курс (11 кл)
Анотація курсу	<p>Даний курс знайомить студентів з поняттями та принципами, які стосуються архітектури та проєктування 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 стекем .NET та реалізацією на його рівні архітектурних принципів та шаблонів.</p>
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=219
Мова викладання	Українська
Лектор курсу	Марченко С.В., спеціаліст I категорії

	<p>канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: sv.marchenko1989@gmail.com</p>
Місце дисципліни в освітній програмі	
Освітня програма	http://csbc.edu.ua/documents/otdel/moop_pr.pdf
Перелік загальних компетентностей (ЗК)	<p>ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК07. Здатність застосовувати знання у практичних ситуаціях.</p>
Перелік спеціальних компетентностей (СК)	<p>СК01. Здатність алгоритмічно та логічно мислити.</p> <p>СК02. Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя.</p> <p>СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.</p> <p>СК04. Здатність дотримуватися стандартів при розробці програмного забезпечення.</p> <p>СК06. Здатність брати участь у проектуванні програмного забезпечення.</p> <p>СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів.</p>
Перелік програмних результатів навчання	<p>РН02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення.</p> <p>РН03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення.</p> <p>РН05. Розробляти та супроводжувати програмне забезпечення.</p> <p>РН07. Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення.</p>

	<p>PH08. Аналізувати вимоги до програмного забезпечення.</p> <p>PH09. Розуміти основні принципи командної роботи при розробці програмного забезпечення.</p> <p>PH10. Обирати та застосовувати ефективні методи оптимізації алгоритмів.</p> <p>PH11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.</p> <p>PH14. Розуміти предметну область, застосовувати знання у професійній діяльності.</p> <p>PH15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.</p>
Опис дисципліни	
Структура навантаження на студента	<p>Загальна кількість годин – 180</p> <p>Кількість кредитів – 6</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p> <p>Кількість годин для самостійної роботи студентів – 120</p> <p>Форма підсумкового контролю – екзамен</p>
Методи навчання	<p>За подачею навчального матеріалу: методи готових знань, дослідницький метод.</p> <p>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.</p>
Зміст дисципліни	
Тема 1. Вступ до архітектури програмного забезпечення	<p>Базові концепції архітектури програмного забезпечення. Основні види діяльності в галузі архітектури програмного забезпечення. Архітектурні структури та стилі. Дані як аспект архітектури. Базові принципи побудови якісного об'єктно-орієнтованого коду. Побудова UML-діаграм засобами PlantUML.</p>

	Візуалізація архітектури за допомогою моделі C4 та PlantUML.
Тема 2. Атрибути якості та архітектурно значущі вимоги	Архітектурно значущі рішення та їх зв'язок з атрибутами якості. Безпека як аспект архітектури. Масштабованість як аспект архітектури. Продуктивність як аспект архітектури. Стійкість (resilience) як аспект архітектури.
Тема 3. Вступ до патернів проектування. Твірні патерни проектування	Патерни проектування: поняття, опис та класифікація. Проблеми, які вирішують патерни проектування. Вибір та використання патерну. Шаблон «Будівельник». Фабричні шаблони: «фабричний метод». та «Абстрактна фабрика» Шаблон «Прототип». Шаблон «Одиночка».
Тема 4. Структурні патерни проектування	Структурні патерни проектування. Шаблон «Адаптер». Шаблон «Міст». Шаблон «Компонувальник». Шаблон «Декоратор». Шаблон «Фасад». Шаблон «Легковаговик». Шаблон «Заступник».
Тема 5. Поведінкові патерни проектування	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». Модульна контрольна робота 1
Тема 6. Принципи та патерни керування залежностями	Поняття зв'язаності (coupling) та зчеплення (cohesion). Принцип інверсії управління та його прояви. Шаблон «Впровадження залежностей». Поняття та принципи реалізації. DI-патерни. DI-антипатерни та запахи коду. Самостійне створення впровадження залежностей. DI-контейнери в .NET.
Тема 7. Шаблони асинхронного програмування	Асинхронне виконання коду. Застарілі моделі асинхронного програмування: Event-based Asynchronous Pattern та Asynchronous Programming Model (APM) pattern. Сучасне

	асинхронне програмування: Task-based Asynchronous Pattern.
Тема 8. Проектування сучасних вебдодатків на базі ASP.NET Core	Характеристики та архітектурні принципи сучасних вебдодатків. Загальні архітектурні стилі вебдодатків. Використання шаблону «впровадження залежностей» в ASP.NET Core. Проміжні (middleware) компоненти ASP.NET Core. Розробка додатків за допомогою ASP.NET Core MVC. Робота з даними в ASP.NET Core додатках. Технологія LINQ. Технологія Entity Framework Core. Поняття та розробка мікросервісів. Процес SEED(S). Вибір оптимального розміру мікросервісів. Робота з даними. Створення конвеєру інфраструктури. Створення інфраструктури мікросервісів. Розробка вебсайту на базі мікросервісної 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.
Система оцінювання	
Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку, відповідно до графіку навчального процесу.	

Підсумкова оцінка за умови заліку виставляється як загальна сума балів, набраних за результатами поточного контролю.

Накопичування рейтингових балів з навчальної дисципліни

Види навчальної роботи		Мах кількість балів
Аудиторна		
Практичні завдання (8 тем)		34
T1	T2	T3
T4	T5	T6
T7	T8	
5	4	3
4	4	4
3	4	3
4	3	7
Тестування		6
Екзамен		30
Індивідуальна		
Завдання до тем		30
T1	T2	T3
T4	T5	T6
T7	T8	
5	10	-
-	-	-
-	-	-
-	-	15
РАЗОМ		100

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

Основна література

1. Sazanavets F. The easiest way to learn design patterns With C# code samples using .NET 6 templates. Lean Publishing, 2022. 326с.

2. Ford N. et al. Building Evolutionary Architectures: Automated Software Governance, 2nd Edition. Sebastopol: O'Reilly Media, 2022. 262p. ISBN: 9781492097549.
3. Erder M., Pureur P., Woods E. Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps. Boston: Addison-Wesley, 2021. 322c.
4. Bass L., Clements P., Kazman R. Software Architecture in Practice (SEI Series in Software Engineering), 4th Edition. Boston: Addison-Wesley, 2021. 442p.
5. 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.
6. 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. 360c.
7. Bell M. Software Architect. Hoboken: Wiley, 2023. 388c.
8. Richards M. Software Architecture Patterns. Understanding Common Architectural Styles and When to Use Them. Sebastopol: O'Reilly Media, 2022. 68p.
9. van Deursen S., Seemann S. Dependency Injection: Principles, Practices, and Patterns. NY, Shelter Island: Manning Publications, 2019. 532c.

Допоміжна література

1. Sarcar V. Design Patterns in C#: A Hands-on Guide with Real-World Examples. NY: Appress, 2018. 456p.
2. 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.
3. Baptista G., Abbruzzese F. Software Architecture with C# 10 and .NET 6. Develop software solutions using microservices, DevOps, EF Core, and design patterns for Azure, Third Edition. Birmingham: Packt Publishing, 2022. 714p.
4. 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.
5. Kumar R., Khan S. A., Khan R. A. Software Durability: Concepts and Practices. Boca Raton: CRC Press, 2023. 330p.
6. 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.

Інформаційні 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/#>.