

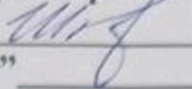
ЗАКЛАД ВИЩОЇ ОСВИТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з методичної роботи

 Ярослав ШТАНЬКО
“ ___ ” _____ 2024 р.

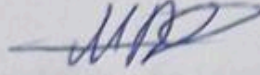
СУЧАСНА АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Інженерія програмного забезпечення
Освітній рівень:	другий (магістерський)
Статус дисципліни:	обов'язкова
Мова викладання, навчання та оцінювання:	українська

РОЗРОБНИК:

к.т.н., доцент кафедри ІТ

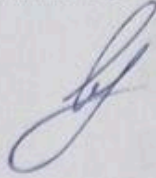


Микола ДЕМЧИНА

ЗАТВЕРДЖЕНО:

на засіданні кафедри інформаційних технологій, протокол № 1 від 28.08.2024 р.

Завідувач кафедри



Сергій ВАЩИШАК

УЗГОДЖЕНО:

Гарант ОПП



Сергій ВАЩИШАК

на засіданні кафедри інформаційних технологій, протокол № 1 від 28.08.2024 р.

Завідувач кафедри



Сергій ВАЩИШАК

СХВАЛЕНО:

на засіданні Науково-методичної ради, протокол № 1 від 30.08.2024 р.

е-mail	mykola.demchyna@ukd.edu.ua
Номер аудиторії чи кафедри	Кафедра інформаційних технологій, ауд. 206
Посилання на сайт	https://ukd.edu.ua
Сторінка курсу в СДО	https://online.ukd.edu.ua

ВСТУП

Анотація навчальної дисципліни

Мета вивчення дисципліни “Сучасна архітектура ПЗ” – надати студентам повного розуміння різноманітних моделей і технологій сучасної архітектури програмного забезпечення. Цей курс дозволить студентам отримати практичний досвід у проектуванні, реалізації та оцінці архітектури програмних додатків у сучасних середовищах розробки.

Основні завдання, що стоять перед авторами курсу, полягають в тому, щоб розкрити здобувачам:

- основні концепції і принципи сучасної архітектури програмного забезпечення;
- провести аналіз різних сучасних шаблонів архітектури програмного забезпечення;
- важливість контейнеризації, оркестровки та доменно-керованого дизайну в сучасній розробці програмного забезпечення;
- практичні навички проектування та реалізації сучасних архітектур програмного забезпечення, використовуючи реальні приклади та тематичні дослідження;
- здатність критично оцінювати архітектури програмного забезпечення та визначати відповідні дизайнерські рішення та компроміси;
- створення детального плану архітектури сучасного програмного забезпечення, включаючи дизайн компонентів, потоку даних і обґрунтування вибору архітектури.

В результаті вивчення дисципліни студент повинен вміти:

- використовувати основні принципи, концепції та практики сучасної архітектури програмного забезпечення;
- розуміти значення архітектурних рішень у забезпеченні масштабованості, продуктивності, безпеки та ремонтпридатності системи;
- виділяти ключові компоненти та принципи проектування різних сучасних архітектурних шаблонів, таких як мікросервіси, безсерверний, та інші;
- розуміти роль платформ контейнеризації та оркестровки, таких як Docker і Kubernetes, в архітектурі розподілених програм;
- застосовувати принципи доменно-орієнтованого проектування для створення гнучких програмних систем, які можна підтримувати;
- використовувати методи оцінювання та порівняння архітектур програмного забезпечення на основі відповідних показників і критеріїв;
- вибирати найкращі практики та міркування, пов'язані з проектуванням, впровадженням та розгортанням сучасних програмних систем.

Компетентності та результати навчання, яких набувають здобувачі освіти внаслідок вивчення навчальної дисципліни “Сучасна архітектура ПЗ” (шифри та зміст компетентностей та програмних результатів навчання вказано відповідно до ОПП/ОНП «Інженерія програмного забезпечення»).

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.	РН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв’язання складних задач інженерії програмного забезпечення.
ЗК05. Здатність генерувати нові ідеї (креативність).	
СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.	
СК03. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.	
СК04. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.	
СК05. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.	
СК06. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами у сфері інженерії програмного забезпечення.	РН08 . Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
	РН10. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	1		
Семестр	2		
Кількість кредитів ECTS	3		
		Денна форма	Заочна форма
Аудиторні навчальні заняття	лекції	14 (в годинах)	4 (в годинах)
	семінари, практичні	16 (в годинах)	4 (в годинах)
Самостійна робота		60 (в годинах)	82 (в годинах)
Форма підсумкового контролю	Екзамен		

СТРУКТУРНО-ЛОГІЧНА СХЕМА ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Пререквізити	Постреквізити
Методологія розробки ПЗ	Кваліфікаційна робота

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Перелік тем лекційного матеріалу

Тема 1. Вступ до сучасної архітектури програмного забезпечення (2 год.)

1. Визначення архітектури програмного забезпечення.
2. Значення архітектури програмного забезпечення в сучасних додатках.
3. Принципи сучасної архітектури програмного забезпечення.
4. Роль архітекторів програмного забезпечення.
5. Огляд типових сучасних шаблонів архітектури програмного забезпечення.

Тема 2. Мікросервісна архітектура (2 год.)

1. Визначення та переваги мікросервісної архітектури.
2. Компоненти мікросервісної архітектури.
3. Зв'язок між мікросервісами.
4. Принципи проектування для створення мікросервісів.

Тема 3. Безсерверна архітектура (2 год.)

1. Визначення та переваги безсерверної архітектури.
2. Функції як послуга (FaaS) і Backend як послуга (BaaS).
3. Загальні випадки використання безсерверної архітектури.
4. Порівняння безсерверної архітектури з архітектурою мікросервісів.

Тема 4. Контейнеризація та оркестризація (2 год.)

1. Вступ до контейнеризації.
2. Переваги використання контейнерів.
3. Основи Docker: створення та керування контейнерами.
4. Вступ до оркестровки контейнера.
5. Основи Kubernetes: керування розгортанням і масштабуванням контейнерів.

Тема 5. Архітектура, керована подіями (2 год.)

1. Принципи керованої подіями архітектури.
2. Переваги та застосування архітектури, керованої подіями.
3. Шаблони обробки подій: прості, потокові та складні.

4. Реалізація керованої подіями архітектури з використанням систем обміну повідомленнями.

Тема 6. Доменно-орієнтований дизайн і обмежений контекст (2 год.)

1. Вступ до доменно-орієнтованого проектування (DDD).
2. Агрегати, об'єкти значень та сутності в DDD.
3. Інтеграція з іншими обмеженими контекстами за допомогою Анतिकорупційного рівня (ACL)

Тема 7. Оцінка архітектури програмного забезпечення (2 год.)

1. Важливість оцінювання архітектури програмного забезпечення.
2. Принципи оцінювання архітектури програмного забезпечення
3. Загальні методи оцінювання: метод аналізу компромісів архітектури (ATAM) і метод аналізу витрат і вигод (CBAM).
4. Метрики для оцінювання архітектур програмного забезпечення

Зміст практичних занять

Тема 1. Виявлення шаблонів архітектури програмного забезпечення в існуючих програмах (2 год.)

Мета: Аналіз існуючої програми та надання загального огляду архітектури. Виявлення архітектурних моделей і міркування щодо вибору дизайну. Знайомство з різними шаблонами архітектури програмного забезпечення.

Тема 2. Проектування архітектури мікросервісів для зразка програми (2 год.)

Мета: Виявлення та визначення меж обслуговування. Проектування зв'язку між мікросервісами. Планування потенційних викликів і проблем впровадження.

Тема 3. Реалізація безсерверної функції та її розгортання в хмарному провайдері (4 год.)

Мета: Написання простої безсерверної функції для вирішення конкретного завдання. Розгортання та тестування безсерверних функцій за допомогою хмарної платформи Functions-as-a-Service (FaaS).

Тема 4. Створення та розгортання контейнерної програми за допомогою Docker і Kubernetes (2 год.)

Мета: Навчитися створювати Dockerfile для програми. Створення та запуск контейнера Docker локально. Розгортання та керування контейнерними програмами за допомогою Kubernetes.

**Тема 5. Проектування системи керованої подіями архітектури
для зразка програми (2 год.)**

Мета: Визначення виробників подій, споживачів і каналів для керованої подіями архітектури. Реалізація обробки подій за допомогою платформи обміну повідомленнями. Забезпечення послідовності та надійності повідомлень.

**Тема 6. Застосування принципів дизайну орієнтованого на домен
до реальної проблеми (2 год.)**

Мета: Проектування моделі предметної області на основі реальної проблеми. Ідентифікація піддоменів і обмежених контекстів. Застосування концепцій дизайну, керованого доменом, таких як агрегати, об'єкти значень та сутності.

Тема 7. Оцінка архітектури зразка програми (2 год.)

Мета: Застосування методу оцінки архітектури до заданої архітектури. Виявлення потенційних архітектурних ризиків і вдосконалень. Пріоритезація архітектурних проблем і компромісів на основі результатів оцінювання. Відгуки студентів про їхній досвід навчання.

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни:

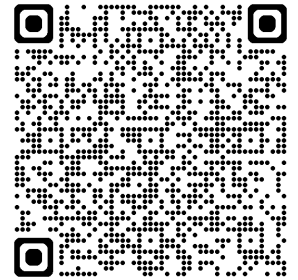
Найменування видів робіт	Розподіл годин	
	денна форма	заочна форма
Самостійна робота, год, у т.ч.:	60	82
Опрацювання матеріалу, викладеного на лекціях	6	8
Підготовка до практичних занять та контрольних заходів	24	10
Підготовка звітів з практичних робіт	20	4
Підготовка до поточного контролю	5	10
Опрацювання матеріалу, винесеного на самостійне вивчення	5	50

ПОЛІТИКА КУРСУ

Коротко, з покликанням на відповідну нормативну базу УКД, висвітлити питання:

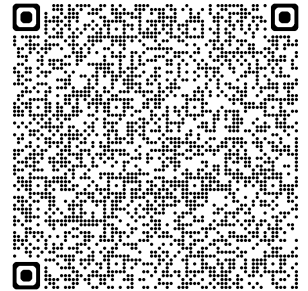
1) щодо системи поточного і підсумкового контролю

Організація поточного та підсумкового семестрового контролю знань студентів, проведення практик та атестації, переведення показників академічної успішності за 100-бальною шкалою в систему оцінок за національною шкалою здійснюється згідно з “Положенням про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти”. Ознайомитись з документом можна за [покликанням](#).



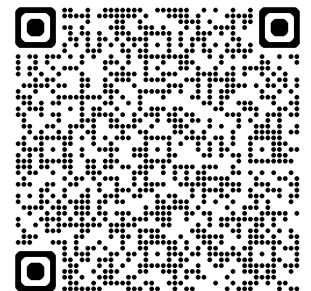
2) щодо оскарження результатів контрольних заходів

Здобувачі вищої освіти мають право на оскарження оцінки з дисципліни отриманої під час контрольних заходів. Апеляція здійснюється відповідно до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за [покликанням](#).



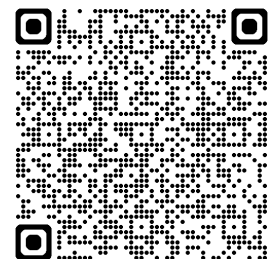
3) щодо відпрацювання пропущених занять

Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку), якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за [покликанням](#).

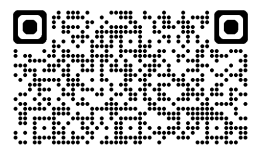


4) щодо дотримання академічної доброчесності

“Положення про академічну доброчесність” закріплює моральні принципи, норми та правила етичної поведінки, позитивного, сприятливого, доброчесного освітнього і наукового середовища, професійної діяльності та професійного спілкування спільноти Університету, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання. Ознайомитись з документом можна за [покликанням](#).



5) щодо використання штучного інтелекту



“Положення про академічну доброчесність” визначає політику щодо використання технічних засобів на основі штучного інтелекту в освітньому процесі. Ознайомитись з документом можна за [посиланням](#).¹ “Положення про систему запобігання та виявлення академічного плагіату, самоплагіату, фабрикації та фальсифікації академічних творів” містить рекомендації щодо використання в академічних текстах генераторів на основі штучного інтелекту. Ознайомитись з документом можна за [посиланням](#).

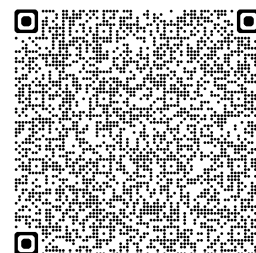
б) щодо використання технічних засобів в аудиторії та правила комунікації

Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання.

Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

7) щодо зарахування результатів навчання, здобутих шляхом формальної/інформальної освіти

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [посиланням](#).²



МЕТОДИ НАВЧАННЯ

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця.

Програмний результат навчання³	Метод навчання	Метод оцінювання
РН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.	словесні методи (лекція, розповідь пояснення); наочні методи (ілюстрування, комп'ютерні і мультимедійні методи, зразки коду, зразки програм);	поточний контроль, екзамен

¹ визначається політика використання ШІ в навчальній дисципліні - дозволене/заборонене, правила використання

² визначається перелік електронних та інших ресурсів та умови перезарахування

³ для вибіркових навчальних дисциплін вказується **результат навчання**

<p>PH08 . Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.</p> <p>PH10. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.</p>	<p>інтерактивні методи (дискусія, мозковий штурм, робота в команді (групах)); метод симулювання проектів;</p>	
--	---	--

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Контрольні заходи

<i>Вид</i>	<i>Зміст</i>	<i>% від загальної оцінки</i>	<i>Бал</i>	
			<i>min</i>	<i>max</i>
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	екзамен	40	25	40
Всього:		100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація **поточного** контролю здійснюється в “Електронному журналі обліку успішності академічної групи” на підставі чотирибальної шкали - “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».

Критерії оцінювання (за необхідності, поточного та/або підсумкового контролю):

<p>«незадовільно»</p>	<p>володіє навчальним матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів, що позначаються окремими словами чи реченнями; володіє матеріалом на елементарному рівні засвоєння, викладає його уривчастими реченнями, виявляє здатність висловити думку на елементарному рівні; володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу;</p>
<p>«задовільно»</p>	<p>володіє матеріалом на початковому рівні, значну частину матеріалу відтворює на репродуктивному рівні; володіє матеріалом на рівні, вищому за початковий, здатний за допомогою викладача логічно відтворити значну його частину; може відтворити значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, за допомогою викладача може аналізувати навчальний матеріал, порівнювати та робити висновки, виправляти допущені помилки;</p>
<p>«добре»</p>	<p>здатний застосовувати вивчений матеріал на рівні стандартних ситуацій, частково контролювати власні навчальні дії, наводити окремі власні приклади на підтвердження певних тверджень: вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати її на практиці, контролювати власну діяльність, виправляти помилки і добирати аргументи на підтвердження певних думок під керівництвом викладача; вільно володіє вивченим обсягом матеріалу, та вміє застосовувати його на практиці; вільно розв'язує задачі в стандартних ситуаціях, самостійно виправляє допущені помилки, добирає переконливі аргументи на підтвердження вивченого матеріалу;</p>
<p>«відмінно»</p>	<p>виявляє початкові творчі здібності, самостійно визначає окремі цілі власної навчальної діяльності, оцінює окремі нові факти, явища, ідеї; знаходить джерела інформації та самостійно використовує їх відповідно до цілей, поставлених викладачем; вільно висловлює власні думки і відчуття, визначає програму особистої пізнавальної діяльності, самостійно оцінює різноманітні життєві явища і факти, виявляючи особисту позицію щодо них; без допомоги викладача знаходить джерела інформації і</p>

	використовує одержані відомості відповідно до мети та завдань власної пізнавальної діяльності; використовує набуті знання і вміння в нестандартних ситуаціях; виявляє особливі творчі здібності, самостійно розвиває власні обдарування і нахили, вміє самостійно здобувати знання.
--	---

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення дисципліни впродовж наступного навчального семестру. За результатами підсумкового контролю (диференційований залік/екзамен) студент може отримати 40 балів. Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен/диференційований залік і повинні йти на перездачу.

Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно з **Шкалою оцінювання знань за ЄКТС**) і є сумою балів отриманих під час поточного та підсумкового контролю.

Шкала оцінювання знань за ЄКТС:

Оцінка за національною шкалою	Рівень досягнень, %	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

Студенти, які не з'явилися на заліки/екзамени без поважних причин, вважаються такими, що одержали незадовільну оцінку.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ⁴

Основна література

1. "Microservices Patterns: With examples in Java - Second Edition" by Chris Richardson, February 2022, Manning Publications
2. "Designing Event-Driven Systems: Concepts and Patterns for Streaming Services with Apache Kafka" by Ben Stopford, June 2021, O'Reilly Media Fowler, Kent Beck, and Erich Gamma (2021).
3. "Serverless Design Patterns and Best Practices" by Brian Zambrano, February 2021, Packt Publishing.
4. "Micro Frontends in Action" by Michael Geers, November 2020, Manning Publications.
5. "Domain-Driven Design: The First 15 Years" by Eric Evans, Vaughn Vernon, Paul Rayner, and other authors, October 2020, Domain Language Inc.
6. "Kubernetes Best Practices: Blueprints for Building Successful Applications on Kubernetes" by Brendan Burns, Eddie Villalba, Lachlan Evenson, and Dave Strelbel, November 2020, O'Reilly Media.
7. "Learning Domain-Driven Design: Unleashing Complexities and Embracing Unknowns" by Vlad Khononov, July 2021, Leanpub.
8. "Implementing Microservices: A Comprehensive Guide to Designing, Building, and Deploying Microservices Architecture" by Hasan Abedi, January 2021, Addison-Wesley Professional

Додаткова література

1. Chen, L., Xia, X., Wang, Q., & Yang, Z. (2020). Continuous Software Quality Assurance Based on Developer Social Network Analysis. *IEEE Transactions on Software Engineering*.
2. García, J., Vivanco, J., & Ebert, C. (2020). Quality Intelligence in Software Development. *IEEE Software*, 37(3), 68-72.

⁴ обов'язково: враховувати вимоги [ДСТУ 8302:2015](#) (відповідно до [Наказу № 65, від 4.03. 2016](#)), [рекомендації](#) Національного агентства з забезпечення якості вищої освіти, використовувати літературу за останні 5-7 років, наводити власні публікації за змістом навчальної дисципліни.