

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»



Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

«29» червня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«ХМАРНІ ТЕХНОЛОГІЇ ТА РОЗПОДІЛЕНЕ ПРОГРАМУВАННЯ»

для здобувачів вищої освіти

галузь знань
спеціальність
освітній рівень
освітня програма
вид дисципліни

12 Інформаційні технології
121 Інженерія програмного забезпечення
бакалавр
Інженерія програмного забезпечення
обов'язкова

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	7
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладач: **ПОРЄВ Геннадій Володимирович, доктор технічних наук, ст. наук. співроб**

Пролонговано: на 2027/2028 н.р. _____ (_____) «____» _____ 2027 р.
на 2028/2029 н.р. _____ (_____) «____» _____ 2028 р.
на 2029/2030 н.р. _____ (_____) «____» _____ 2029 р.

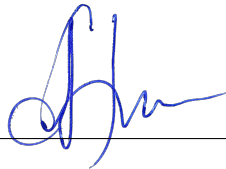
Розробники:

ПОРЄВ Геннадій Володимирович, доктор технічних наук, старший науковий співробітник, доцент кафедри програмних систем і технологій



«ЗАТВЕРДЖУЮ»

Завідувач кафедри
програмних систем і технологій



проф Олексій БИЧКОВ

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року



Голова науково-методичної комісії _____ Ганна КРАСОВСЬКА

1. Мета дисципліни

Полягає у формуванні в здобувачів освіти системних знань і практичних навичок з проєктування, розгортання та супроводу хмарних і розподілених програмних систем. Особливий акцент робиться на сучасні підходи до контейнеризації та оркестрації, моделі узгодженості даних, міжсервісну взаємодію (синхронну й асинхронну), забезпечення відмовостійкості, спостережуваності та безпеки, а також на застосування інфраструктури як коду та CI/CD практик для побудови cloud-native рішень.

2. Попередні вимоги до опанування або вибору навчальної дисципліни

Базові знання з:

- структур і алгоритмів, парадигм програмування; практичний досвід програмування (бажано мовою Go; допускаються інші мови загального призначення);
- операційних систем і комп'ютерних мереж (модель OSI/TCP/IP, DNS, HTTP/HTTPS);
- баз даних (моделі даних, транзакції, індекси; базове SQL);
- роботи з ОС Linux (командний рядок, пакетні менеджери, робота з файлами та процесами);
- Git та основи керування версіями.

Бажано: елементи архітектур ПЗ, основи веб-технологій, знайомство з контейнерами (Docker/Podman).

3. Анотація навчальної дисципліни

Дисципліна є складовою циклу професійної підготовки здобувачів освітнього рівня «бакалавр» за спеціальністю «121 Інженерія програмного забезпечення». Курс охоплює моделі надання хмарних послуг (IaaS/PaaS/SaaS), контейнери та їх оркестрацію, побудову мікросервісної архітектури, синхронну (REST/gRPC) та асинхронну (черги/шини подій) взаємодію, моделі узгодженості даних, масштабування та відмовостійкість, спостережуваність (логування, метрики, трасування), безпеку та економіку хмарних обчислень. Практична частина зосереджена на створенні та розгортанні cloud-native застосунків з використанням інфраструктури як коду і безперервної інтеграції/доставки. Методи викладання: лекції, лабораторні заняття, case study, самостійна робота здобувачів. Методи оцінювання: звіти з лабораторних робіт, тестування/усні опитування, контрольні роботи за змістовими модулями, підсумковий іспит.

4. Завдання (навчальні цілі)

Інтегральна компетентність. Здатність розв'язувати складні задачі і проблеми інженерії програмного забезпечення, що передбачає проведення досліджень з елементами наукової новизни та здійснення інновацій в умовах невизначеності вимог.

Загальні компетентності.

- ЗК-2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК-5. Здатність вчитися та оволодівати сучасними знаннями та проводити самостійні дослідження на відповідному рівні.
- ЗК-6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Фахові компетентності спеціальності.

- ФК-2. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

- ФК-3. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- ФК-4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.
- ФК-12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.
- ФК-13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- ФК-14. Здатність до алгоритмічного та логічного мислення.

5. Результати навчання за дисципліною

Категорії результатів навчання:

1. знати
2. вміти
3. комунікувати
4. автономність та відповідальність

Методи навчання:

- **словесний:** лекція, дискусія, співбесіда тощо
- **наочний:** ілюстрацій і метод демонстрацій
- **практичний:** лабораторні заняття, розрахункові роботи
- **відеометод:** у сполученні з інформаційними технологіями та комп'ютерними засобами навчання (дистанційні, мультимедійні, веб-орієнтовані тощо)
- **література:** робота з навчально-методичною літературою та іншими джерелами інформації (конспектування, тезування, анотування, складання реферату, есе)
- **самостійна:** робота по індивідуальним творчим завданням, розв'язання програмних завдань: домашні та лабораторні

Код	Результат навчання	Методи викладання і навчання	Методи оцінювання	Відсоток
1.1	Знати моделі та принципи хмарних обчислень, основні підходи до контейнеризації та оркестрації програмних систем	словесний, наочний, практичний, відео-метод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
1.2	Знати основні моделі надання хмарних послуг та способи їх застосування для розгортання програмних систем	словесний, наочний, практичний, відео-метод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%

1.3	Знати принципи побудови cloud-native систем, забезпечення їх масштабованості, відмовостійкості та доступності.	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
1.4	Знати принципи організації взаємодії між сервісами у розподілених та контейнеризованих системах	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
1.5	Знати сучасні підходи до моніторингу, забезпечення безпеки, автоматизації розгортання та керування інфраструктурою хмарних систем	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
2.1	Вміти аналізувати вимоги до програмної системи та обирати архітектурні рішення з урахуванням особливостей хмарного середовища	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	8%
2.2	Вміти обґрунтовано обирати моделі та технології розподіленого програмування для реалізації хмарних сервісів	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	8%
2.3	Вміти проектувати cloud-native сервіси з урахуванням вимог щодо масштабованості, відмовостійкості та доступності	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
2.4	Вміти реалізовувати взаємодію між сервісами та розгортати програмні компоненти у контейнеризованому середовищі	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
2.5	Вміти налаштовувати моніторинг, безпеку, інфраструктуру як код та засоби CI/CD для розподілених програмних систем	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
3.1	Вміти ефективно взаємодіяти з учасниками команди під час проектування, розроблення та розгортання хмарних програмних систем	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
3.2	Вміти документувати архітектурні рішення, конфігурацію та результати розгортання хмарних систем і презентувати їх зацікавленим сторонам	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%

4.1	Вміти самостійно організувати процес розгортання та супроводження програмних систем у хмарному середовищі в межах поставленого завдання	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
4.2	Вміти обґрунтовувати вибір хмарних технологій, архітектурних рішень та засобів автоматизації з урахуванням технічних і ресурсних обмежень	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%
4.3	Нести відповідальність за надійність, безпеку та коректність виконання процесів розгортання і супроводження хмарних програмних систем	словесний, наочний, практичний, відеометод, література, самостійна;	тестування, опитування, модульні контрольні роботи, захист лабораторних робіт, іспит;	7%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Програмні результати навчання

- ПР-1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
- ПР-5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.
- ПР-6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.
- ПР-14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.
- ПР-28. Знати архітектуру програмного забезпечення та принципи проєктування, володіти технологіями та методологіями розробки та реалізації програмного забезпечення та мережними технологіями, мати практичні навички з адміністрування комп'ютерних мереж в процесі розробки, розгортання та супроводження програмного забезпечення, розуміти і застосовувати програмне забезпечення мережевого обміну, обміну з сенсорами, віддаленої та розподіленої обробки інформації в інтересах інтернет речей.
- ПР-29. Знати, розуміти, аналізувати, вибирати алгоритми та програмні засоби машинного навчання і аналізу даних в процесі розробки програмного забезпечення для завдань професійної діяльності, у.ч. в процесі маркетингових досліджень, а також вміти реалізовувати розподілені обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень.

Результат навчання	ПРН1	ПРН5	ПРН6	ПРН14	ПРН28	ПРН29
1.1	+		+		+	+
1.2	+					+
1.3		+			+	+
1.4					+	+
1.5				+	+	+
2.1	+		+		+	

2.2	+		+			+
2.3		+			+	+
2.4				+	+	+
2.5			+	+	+	+
3.1			+	+		
3.2	+			+	+	
4.1			+	+	+	+
4.2	+		+		+	+
4.3				+	+	+

7. Схема формування оцінки

7.1. Форми оцінювання здобувачів вищої освіти:

Контроль знань здобувачів вищої освіти здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності здобувачів вищої освіти оцінюються за 100-бальною шкалою. Робота в семестрі поділяється на два змістових модуля. Підсумковий контроль (залік) – 40 балів. Обов'язковим для допуску до заліку є виконання та захист здобувачем вищої освіти усіх лабораторних робіт, що передбачено робочою програмою навчальної дисципліни. Під час цього сумарна мінімальна рекомендована кількість балів для допуску 36.

	ЗМ1/Частина 1	ЗМ2/Частина 2	залік	Підсумкова оцінка
<i>Мінімум</i>	18	18	24	60
Максимум	30	30	40	100

До поточної підсумкової оцінки викладач може додати заохочувальні бали, які можна отримати активною роботою на лекціях, лабораторних заняттях, за виконання самостійних проєктів, участі в конференції з доповіддю за тематикою дисципліни. Заохочувальні бали можуть бути нараховані за проходження сертифікованих курсів з розробки та розгортки програмного забезпечення в хмарних середовищах (до 5 балів). Максимальна кількість заохочувальних балів за семестр – 10 балів, які є складовою поточного оцінювання.

Оцінка за екзамен не може бути меншою 24 балів для отримання загальної позитивної оцінки за курс. Якщо здобувач на екзамені набрав менше 24 балів (60% від 40 балів, відведених на екзамен), то вони не додаються до семестрової оцінки незалежно від кількості балів, отриманих під час семестру, а в екзаменаційній відомості у графі «результуюча оцінка» переноситься лише кількість балів, отриманих під час семестру.

7.2. Організація оцінювання

Оцінювання проводиться відповідно до «Рекомендацій щодо оцінювання освітніх STEM-проєктів», затверджених на засіданні кафедри програмних систем і технологій (протокол № 16 від 1 травня 2026 р.): <https://pst.knu.ua/academics/bachelor/evaluation-stem-projects/>

Поточне семестрове оцінювання

- оцінюванню підлягає модульна контрольна робота, звіти та захист лабораторних робіт, активна робота на лекціях і лабораторних заняттях;
- впродовж семестру за змістовними модулями 1 та 2 проводяться модульні контрольні роботи на лабораторних або практичних заняттях (здобувачі вищої освіти отримують завдання за варіантами та виконують їх письмово). Максимальна кількість балів за кожну з модульних контрольних робіт не перевищує 6, модульні контрольні роботи не перескладаються. Перевірка завдань, оприлюднення та аналіз результатів здійснюється не пізніше наступного лабораторного заняття;

- кожна лабораторна робота оцінюється в 6 балів, якщо всі завдання виконано, продемонстровано їх працездатність, звіт оформлено відповідно до вимог. Максимальна кількість балів за усі лабораторні роботи не перевищує 48. Кінцевим терміном здачі кожної лабораторної роботи є наступне лабораторне заняття після виконання зазначеної лабораторної роботи;
- для допуску до екзамену обов'язковим є виконання та захист усіх запланованих лабораторних робіт, критично-розрахунковий мінімум для допуску до заліку становить 36 балів.

Підсумкове оцінювання (у формі заліку):

- екзаменаційний білет складається з тестової та практичної частин (виконуються письмово);
- максимальна кількість балів, які можуть бути отримані здобувачем вищої освіти на екзамені
- 40 балів;
- здобувач не допускається до заліку, якщо під час семестру набрав менше ніж 36 балів;
- для отримання загальної позитивної оцінки з дисципліни кількість балів для розрахунку результату заліку не може бути меншою 24 балів.

7.3. Шкала відповідності

Зараховано/Pass	60—100
Незараховано/Fail	0-59

8. Структура навчальної дисципліни

7 семестр.

М.№	Назва теми	Лек. год.	Лаб. год.	С/Р. год.
1.1	Вступ. Моделі хмарних послуг (IaaS/PaaS/SaaS), моделі розгортання, shared-responsibility	2	2	4
1.2	Віртуалізація та контейнеризація: VM vs Containers, образи, реєстри, базові практики	2	3	6
1.3	Хмарні мережі: віртуальні мережі, маршрутизація, балансування, service discovery	2	2	4
1.4	Сховища і дані: об'єктне/блочне/файлове зберігання; реляційні та NoSQL БД у хмарі	2	3	6
1.5	Мікросервісна архітектура. Синхронні API: REST/gRPC, API-gateway, контрактне тестування	2	2	4
1.6	Асинхронна взаємодія: черги та стріми (чергові брокери/шини подій), Pub/Sub, ідемпотентність	3	3	6
1	Модульна контрольна робота 1	2		
2.1	Моделі узгодженості (CAP, PACELC), клади та таймаути, ретраї, дедуплікація	2	2	4
2.2	Відмовостійкість і надійність: реплікація, шардінг, патерни (Circuit Breaker, Bulkhead), chaos-тестування	2	3	6

2.3	Оркестрація контейнерів: Kubernetes (Pods, Deployments, Services, Ingress), конфігурації/секрети	2	2	4
2.4	Інфраструктура як код: Terraform (базові поняття), CI/CD, огляд GitOps	2	3	6
2.5	Спостережуваність: логування, метрики, трасування (OpenTelemetry), SLI/SLO/Error Budget	2	2	4
2.6	Безпека й економіка хмари: IAM, шифрування, керування секретами; Serverless/FaaS та edge-підходи	3	3	6
2	Модульна контрольна робота 2	2		
	Разом	30	30	60

Всього: лекції — 30 год, лабораторні — 30 год, самостійна робота — 60 год.
Загальний обсяг: 120 год.

9. Рекомендована література

9.1. Основна література

1. Kleppmann M. Designing Data-Intensive Applications. O'Reilly, 2017. — Фундамент про моделі даних, узгодженість, події та потоки.
2. Tanenbaum A. S., van Steen M. Distributed Systems, 3rd ed. Pearson, 2017. — Класика принципів розподілених систем.
3. Newman S. Building Microservices, 2nd ed. O'Reilly, 2021. — Архітектура мікросервісів, інтеграція та еволюція.
4. Nygard M. T. Release It! 2nd ed. Addison-Wesley, 2017. — Патерни надійності, стійкість у продакшені.
5. Beyer B., Jones C., Petoff J., Murphy N. R. (eds.) Site Reliability Engineering. O'Reilly, 2016; Beyer B. та ін. The Site Reliability Workbook. O'Reilly, 2018. — Принципи SRE, SLI/SLO, експлуатація.
6. Hightower K., Burns B., Beda J. Kubernetes: Up & Running, 3rd ed. O'Reilly, 2023. — Практичне введення до Kubernetes.
7. Brikman Y. Terraform: Up & Running, 3rd ed. O'Reilly, 2022. — Інфраструктура як код і практики IaC.
8. Burns B. Designing Distributed Systems. O'Reilly, 2018. — Шаблони побудови розподілених систем на контейнерах.
9. Richardson C. Microservices Patterns. Manning, 2018. — Патерни для сервісів, транзакційність і Saga.
10. Shapira G., Palino T., Ritchie I. Kafka: The Definitive Guide, 2nd ed. O'Reilly, 2021. — Подієві стріми і платформи даних.

9.2. Додаткова література

1. Indrasiri K., Kuruppu D. gRPC: Up and Running. O'Reilly, 2020. — Високопродуктивні RPC та схемоконтракти.
2. Hausenblas M., Schimanski S. Programming Kubernetes. O'Reilly, 2019. — Розширення та оператори.

3. Barroso L. A., Clidaras J., Hölzle U. The Datacenter as a Computer, 3rd ed. Morgan & Claypool, 2018. — Архітектура дата-центрів та ефективність.
4. Stopford B. Designing Event-Driven Systems. O'Reilly, 2018. — Подієва архітектура та CQRS.
5. Takada M. Distributed Systems for Fun and Profit, 2013. — Короткий огляд основ (вільно доступна електронна версія).

9.3. Онлайн-ресурси та офіційна документація

1. Kubernetes Documentation — концепції, API, найкращі практики.
2. Docker Docs — образи, мережі, безпека контейнерів.
3. OpenTelemetry Docs — трасування, метрики, логування, інструментація.
4. Prometheus та Grafana Docs — метрики, алертинг, візуалізація.
5. gRPC Documentation — протоколи, IDL, генерація клієнтів/серверів.
6. HashiCorp Terraform Docs — провайдери, модулі, стейт, політики.
7. Apache Kafka Documentation — топіки, партиції, схеми, стріми.
8. AWS Well-Architected Framework / Google Cloud Architecture Framework / Azure
9. Well-Architected — перевірені практики проєктування у хмарі.