



Розроблення проблемно-орієнтованих та сервісно-орієнтованих систем

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти

Другий (магістерський)

Галузь знань	F Інформаційні технології
Спеціальність	F2 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	5 кредитів /150 год. Лекцій 30 годин, Практичні -30 годин, Самостійна робота студента - 90 годин
Семестровий контроль/ контрольні заходи	Екзамен
Розклад занять	Згідно розкладу на осінній семестр поточного навчального року за адресою https://schedule.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Керівник (забезпечує курс): Волокита Артем Миколайович, доцент кафедри обчислювальної техніки, к.т.н., доцент Старший викладач кафедри обчислювальної техніки Шемсєдинов Т.Г. timur.shemsedinov@gmail.com
Розміщення курсу	На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/ODM3Niq4MzE3ODM0?cjc=rhehd5dq

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою навчальної дисципліни є формування у студентів здатностей (компетентностей) розробки програмного забезпечення для проблемно-орієнтованих (DSL) та сервісно-орієнтованих (SOA) систем. За результатами вивчення дисципліни відповідно освітній програмі здобувач має набутти компетентність:

- Здатність розробляти проблемно-орієнтовані та сервісно-орієнтовані системи (ФК14)
- Здатність використовувати мікросервісний підхід для створення програмних систем (ФК15)

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- Знати і застосовувати методи і технології створення проблемно-орієнтованих та сервісно-орієнтованих систем (ПРН25).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити: базові знання з проєктування архітектури програмного забезпечення.

Постреквізити – освітні компоненти: «Методологія інженерії програмного забезпечення», «Науково-дослідна практика», «Виконання магістерської дисертації».

3. Зміст навчальної дисципліни

Тема 1. Класифікація інформаційних систем: ERP, SCM, CRM, MRP, BI, B2B та B2C.

Тема 2. Багатошарова та багатоланкова архітектури для монолітних ІС.

Тема 3. Сервісний підхід: програмні сервіси в середині додатку, веб-сервіси (SOA) та мікросервіси.

Тема 4. Двох та трьох-ланкові додатки, багатоланкові, пайплайн топологія.

Тема 5. Інверсія управління та впровадження залежностей у ІС.

Тема 6. Прозорість для посилань (Referential Transparency) для об'єктно-орієнтованого програмування, функціонального програмування та інших парадигм.

Тема 7. Оркестрація (Orchestration) та Хореографія (Choreography), SAGA Pattern.

Тема 8. Планувальники відкладених завдань та ресурсів, черги виконання.

Тема 9. Тестування, оцінки якості, безперервна інтеграція.

Тема 10. Інфраструктура, розгортання, оновлення, міграції, реінжиніринг.

Тема 11. Балансування, реплікації, шардинг, рещардинг, бекапи та відновлення.

Тема 12. Безпека, авторизація, автентифікація, прикладний фаєрвол.

Тема 13. Прикладне та системне логування, розслідування інцидентів.

Тема 14. Аналіз та реінжиніринг бізнес-процесів.

Тема 15. Serverless Clouds (FaaS) та ізоляція контекстів запитів.

Тема 16. Проєктування API інтерфейсів та контрактів взаємодії систем.

Тема 17. Кодогенерація, шаблони, використання узагальненого програмування для зміни поведінки.

4. Навчальні матеріали та ресурси

Базова література:

1. Матеріали дистанційного курсу – розміщено за адресою:
<https://classroom.google.com/c/ODM3Njg4MzE3ODM0?cjc=rhehd5dq>
2. Шемсєдинов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Приклади коду та приклади проєктів [Електронний ресурс] розміщено за адресою:
<https://github.com/HowProgrammingWorks/>

Додаткова література:

1. Шемсєдинов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Технологічний стек Metarhia [Електронний ресурс] розміщено за адресою: <https://github.com/metarhia/>
2. <https://github.com/HowProgrammingWorks/Index/blob/master/Courses/Architecture-2025.md>
3. Роберт Мартін. Чистий Agile. Переклад укр. Видавництво Фабула, 2021, 224 с. ISBN: 978-617-09-6760-2
4. Martin Fowler et al. Patterns of Enterprise Application Architecture. Addison Wesley, 2002, 560 p. ISBN 0-321-12742-0

5. Steve McConnell. More Effective Agile: A Roadmap for Software Leaders. Construx Press, 2019, 380 p. ISBN: 978-1-7335-1821-5
6. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, Ed. 2000, 368 p.
7. Eric Evans. Domain-Driven Design Reference: Definitions and Pattern Summaries. Domain language, 2015, 59 p.
https://www.domainlanguage.com/wp-content/uploads/2016/05/DDD_Reference_2015-03.pdf
8. Steve McConnell. Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft Press, 2006, 308 p. ISBN: 978-0-7356-0535-0
9. Steve McConnell. Professional Software Development: Shorter Schedules, Higher Quality Products, More Successful Projects, Enhanced Careers. Addison Wesley, 2003, 272 p. ISBN: 9780321193674
10. Robert C. Martin. Clean Agile: Back to Basics. Longman, 2019, 224p. ISBN: 9786170967602
11. Robert C. Martin and Micah Martin. Agile Principles, Patterns, and Practices in C#. Prentice Hall, 2006, 768p. ISBN: 978-0-13-185725-4
12. Robert C. Martin. Agile Software Development, Principles, Patterns, and Practices. 1st ed., Pearson, 2002 [Online] Available from:
<https://dl.ebooksworld.ir/motoman/Pearson.Agile.Software.Development.Principles.Patterns.and.Practices.www.EBooksWorld.ir.pdf>
13. Frederick Brooks Jr. The Mythical Man-Month: Essays on Software Engineering. Addison Wesley, 1995, 336 p. ISBN 0-201-83595-9. [Online] Available from:
<https://www.cesarkallas.net/arquivos/livros/informatica/Addison.Wesley.The.Mythical.Man-Month.Essays.on.Software.Engineering.20th.Anniversary.Edition.pdf>
14. Kent Beck, Cynthia Andres. Extreme Programming Explained. 2nd Edition. Addison Wesley, 2004, 224 p. ISBN: 978-0321278654

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ заняття	Тематика заняття	Ауд. годин.
1	Тема 1. Класифікація інформаційних систем: ERP, SCM, CRM, MRP, BI, B2B та B2C.	2
	Тема 2. Багатошарова та багатоланкова архітектури для монолітних ІС.	
2	Тема 3. Сервісний підхід: програмні сервіси в середині додатку, веб-сервіси (SOA) та мікросервіси.	2
3	Тема 4. Двох та трьох-ланкові додатки, багатоланкові, пайплайн топологія.	2
4	Тема 5. Інверсія управління та впровадження залежностей у ІС.	2
	Тема 6. Прозорість для посилань (Referential Transparency) для об'єктно-орієнтованого програмування, функціонального програмування та інших парадигм.	
5	Тема 7. Оркестрація (Orchestration) та Хореографія (Choreography), SAGA Pattern.	2
6	Тема 8. Планувальники відкладених завдань та ресурсів, черги виконання.	2
7	Тема 9. Тестування, оцінки якості, безперервна інтеграція.	2
8	Тема 10. Інфраструктура, розгортання, оновлення, міграції, реінжиніринг.	2
9	Тема 11. Балансування, реплікації, шардинг, решардинг, бекапи та відновлення.	2

10	Тема 12. Безпека, авторизація, автентифікація, прикладний фаєрвол.	2
11	Тема 13. Прикладне та системне логування, розслідування інцидентів.	2
12	Тема 14. Аналіз та реінжиніринг бізнес-процесів.	2
13	Тема 15. Serverless Clouds (FaaS) та ізоляція контекстів запитів.	2
14	Тема 16. Проектування API інтерфейсів та контрактів взаємодії систем.	2
	Тема 17. Кодогенерація, шаблони, використання узагальненого програмування для зміни поведінки.	
15	Модульна контрольна робота	2
	Всього:	30

Практичні заняття

Основні завдання циклу практичних занять (комп'ютерного практикуму) - надбання практичних навичок створення формальних DSL (domain specific language) мов та опису метаданих для динамічного формування бізнес-процесів та доменної моделі, проектування інформаційних систем у архітектурі сервісів SOA (service-oriented architecture), використання DSL мов у SOA системах, використання готових інфраструктур для інтерпретації та виконання вбудованих DSL мов у розподілених сервісах.

На практичних заняттях студенти виконують завдання 6 практичних робіт, теми яких наведені у таблиці

Практичні роботи

№ з/п	Теми практичних робіт	Кількість ауд. годин
1	Розробка архітектури інформаційної системи на базі сервісно-орієнтованої парадигми.	4
2	Аналіз предметної області (на свій вибір) з виділенням метаданих, які можна описати за допомогою DSL мови.	2
3	Проектування DSL мови для опису певної предметної області та створення прикладів використання мови.	6
4	Залучення готових синтаксичних аналізаторів, парсерів, інтерпретаторів, компіляторів, лінтерів та інших засобів розробки для обробки моделей у форматі DSL.	6
5	Використання вбудованих віртуальних машин, рантаймів, середовищ виконання у IC на базі SOA.	6
6	Використання SQL та noSQL СКБД для розподілених SC на базі SOA та DSL.	6
	Всього:	30

6. Самостійна робота студента

У процесі виконання індивідуальних завдань студенти повинні опрацьовувати знання, отримані під час лекцій та самостійної роботи, самостійно вивчати визначені питання, поглиблювати свої знання для подальшого навчання. Орієнтовний розподіл обсягів для окремих видів самостійної наведений у таблиці

Розподіл обсягів самостійної роботи за семестр

№ з/п	Види самостійної роботи	Кількість годин СРС
1	Підготовка до лекцій	8

2	Виконання практичних робіт з вивченням теорії та реалізацією наведеної теми у програмному коді та підготовка доповіді.	30
3	Вивчення питання: Створення SOA систем на базі хмарних FaaS сервісів.	6
4	Вивчення питання: Порівняння хмарних вендорів на предмет ізоляції контекстів при виконанні DSL мов.	4
5	Вивчення питання: Захист IC на базі SOA та DSL від несанкціонованого втручання.	6
6	Підготовка до МКР	6
7	Підготовка до екзамену	30
Всього:		90

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profdkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Порядок визнання таких результатів регламентується Положенням про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>). Можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних завдань, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання.

Політика щодо академічної доброчесності. Обов'язковою умовою виконання завдань з освітньої компоненти є дотримання політики та принципів академічної доброчесності (<https://kpi.ua/academic-integrity>), які, у тому числі, викладено у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>), Положенні про систему запобігання академічному плагіату в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/47>). У разі виявлення дублювання робіт, плагіату роботи здобувачі отримують нульовий рейтинг.

Політика використання штучного інтелекту. Використання штучного інтелекту (далі, ШІ) регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі навчальні завдання з дисципліни мають бути результатом власної оригінальної роботи здобувача.

Організація навчальної роботи. Основним середовищем для зберігання активів програмного коду є репозиторій Github з дотриманням правил OpenSource.

Під час занять з навчальної дисципліни студенти повинні дотримуватись певних дисциплінарних правил:

- 1) так використовувати чат-групи, репозиторії та середовище виконання, щоб не створювати проблем з зайвими нотифікаціями іншим учасникам учбового процесу та викладачеві;
- 2) вчасно виконувати роботу та комітити все кожного дня у систему контролю версій, не створювати багато комітів у одному пул-реквесту, не створювати великих комітів, розділяти все

на окремі тематичні комміти, чітко та зрозуміло описувати комміти та пул-реквести, додавати теги для лінкування issues, PR, та коммітів у репозиторіях;

3) не допускаються використання піратських копій середовищ розробки, операційних систем чи інших засобів розробки та розгортання як на своїх обчислювальних засобах, так і на хмарних;

4) спамити, постити забагато мемів та стікерів у групах та репозиторіях;

5) якщо виникає питання, потрібно спочатку шукати у матеріалах курсу, потім у інтернеті;

Практичні роботи здаються тільки через Github у відкритому коді (до коду студенти додають ліцензію MIT), а репозиторій повинен мати історію розробки, щоб викладач міг простежити послідовність написання коду та авторство. Розробка відбувається у feature-гілках git, після чого перевірка коду проходить через PR та включає рев'ю, яке викладач робить у пул-реквесті. Історія рев'ю залишається у Github аккаунті студента.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Види контролю з навчальної дисципліни «Розроблення проблемно-орієнтованих та сервісно-орієнтованих систем» включають:

Календарний контроль

Перший календарний контроль – на 6-7-му тижні (за графіком). Для отримання «атестовано» потрібно мати поточний рейтинг не менше 5 балів.

Другий календарний контроль – на 13-14-му тижні (за графіком). Для отримання «атестовано» потрібно мати поточний рейтинг не менше 25 балів.

Поточний контроль

Доповіді щодо виконання завдань 6 практичних робіт.

Критерії оцінювання виконання завдань практичних робіт:

- повна відповідь, не менше 90% потрібної інформації – 5 балів;
- достатньо повна відповідь, не менше 75% потрібної інформації, незначні неточності – 4 бали;
- неповна відповідь, не менше 60% потрібної інформації, значні неточності у відповіді по суті питання – 3 бали;
- у відповіді менше 60% потрібної інформації, суттєві неточності – 0-2 балів.

Модульна контрольна робота у вигляді рев'ю кода за курс, які повністю охоплюють розглянуту на лекціях тематику навчальної дисципліни. До рев'ю коду викладач може додавати теоретичні питання, які розкривають розуміння теми студентом.

Критерії оцінювання виконання завдань МКР:

- повна відповідь, не менше 90% потрібної інформації – 18-20 балів;
- достатньо повна відповідь, не менше 75% потрібної інформації, незначні неточності – 15-17 балів;
- неповна відповідь, не менше 60% потрібної інформації, значні неточності у відповіді по суті питання – 12-14 бали;
- у відповіді менше 60% потрібної інформації, суттєві неточності – 0-11 балів.

Семестровий контроль - екзамен

Екзамен проводиться усно, здобувач має відповісти на 5 запитань. Відповідь на кожне запитання по 10-бальній шкалі. Перелік екзаменаційних питань наведений у додатку нижче.

Критерії оцінювання відповідей на запитання екзамену:

- повна відповідь, не менше 90% потрібної інформації – 9-10 балів;
- достатньо повна відповідь, не менше 75% потрібної інформації, незначні неточності розуміння суті питання – 8 балів;
- неповна відповідь, не менше 60% потрібної інформації, значні неточності у відповіді по суті питання – 6-7 балів;

- у відповіді менше 60% потрібної інформації, відповідь не відповідає умовам до «достатньо» – 0-5 балів.

Рейтинг студента з кредитного модуля складається з балів, які він отримує за види робіт відповідно таблиці

Оцінювання окремих видів навчальної роботи студента

Вид навчальної роботи	Максимальна кількість балів
Виконання завдань практичної роботи № 1	5
Виконання завдань практичної роботи № 2	5
Виконання завдань практичної роботи № 3	5
Виконання завдань практичної роботи № 4	5
Виконання завдань практичної роботи № 5	5
Виконання завдань практичної роботи № 6	5
Модульна контрольна робота	20
Екзамен	50
Всього:	100

Умова допуску до екзамену: виконання усіх завдань практичних робіт та МКР, та семестровий рейтинг не менше 40. Сума семестрової та екзаменаційної рейтингових оцінок у балах становить підсумкову семестрову рейтингову оцінку, яка перераховується в оцінки за національною шкалою та шкалою ECTS, як наведено у таблиці

Відповідність рейтингових балів оцінкам за університетською шкалою

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Склали: доцент кафедри обчислювальної техніки, к.т.н., доц. Волокита А.М.,
старший викладач кафедри обчислювальної техніки Шемсєдинов Т.Г.

Ухвалено кафедрою обчислювальної техніки(протокол №12 від 23.06.2025р.)

Погоджено Методичною комісією факультету (протокол №11 від 27.06.2025р.)

Перелік питань, що виносяться на екзамен

1. Розробка архітектури інформаційної системи на базі сервісно-орієнтованої парадигми.
2. Класифікація інформаційних систем: ERP, SCM, CRM, MRP, BI, B2B та B2C.
3. Багатошарова та багатоланкова архітектури для монолітних ІС.
4. Сервісний підхід: програмні сервіси в середині додатку, веб-сервіси (SOA) та мікросервіси.
5. Відмінності сервісно-орієнтованих та монолітних архітектур
6. Двох та трьох-ланкові додатки, багатоланкові, пайплайн топологія.
7. Інверсія управління та впровадження залежностей у ІС.
8. Прозорість для посилань (Referential Transparency) для об'єктно-орієнтованого програмування, функціонального програмування та інших парадигм.
9. Оркестрація (Orchestration) та Хореографія (Choreography), SAGA Pattern.
10. Планувальники відкладених завдань та ресурсів, черги виконання.
11. Тестування, оцінки якості, безперервна інтеграція.
12. Інфраструктура, розгортання, оновлення, міграції, реінжиніринг.
13. Балансування, реплікації, шардинг, рещардинг, бекапи та відновлення.
14. Безпека, авторизація, автентифікація, прикладний фаєрвол.
15. Прикладне та системне логування, розслідування інцидентів.
16. Аналіз та реінжиніринг бізнес-процесів.
17. Serverless Clouds (FaaS) та ізоляція контекстів запитів.
18. Проектування API інтерфейсів та контрактів взаємодії систем.
19. Кодогенерація, шаблони, використання узагальненого програмування для зміни поведінки.
20. Аналіз предметної області (на свій вибір) з виділенням метаданих, які можна описати за допомогою DSL мови.
21. Проектування DSL мови для опису певної предметної області та створення прикладів використання мови.
22. Залучення готових синтаксичних аналізаторів, парсерів, інтерпретаторів, компіляторів, лінерів та інших засобів розробки для обробки моделей у форматі DSL.
23. Використання вбудованих віртуальних машин, рантаймів, середовищ виконання у ІС на базі SOA.
24. Використання SQL та noSQL СКБД для розподілених SC на базі SOA та DSL.

Примітка. Наведений перелік є орієнтовним переліком тематики дисципліни для формування екзаменаційних запитань. На екзамені можуть бути задані більш конкретизовані запитання відповідно наданій тематиці.