



Розроблення проблемно-орієнтованих та сервісно-орієнтованих систем. Курсова робота

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F2 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення комп'ютерних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форманавчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, осінній</i>
Обсяг дисципліни	<i>1 кредит / СРС 30 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року за адресою https://schedule.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу/ викладачів	Керівник (забезпечує курс): Волокита Артем Миколайович, доцент кафедри обчислювальної техніки, к.т.н., доцент, artem.volokita@kpi.ua Старший викладач кафедри обчислювальної техніки Шемсєдинов Тимур Гафарович, timur.shemsedinov@gmail.com .
Розміщення курсу	<i>На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/ODM3Njg4MzE3ODM0?cjc=rhehd5dq</i>

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 здобувач має набути:

- Здатність розробляти проблемно-орієнтовані та сервісно-орієнтовані системи (ФК14),
 - Здатність використовувати мікросервісний підхід для створення програмних систем (ФК15),
- а також продемонструвати 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: Дана дисципліна є підготовкою до виконання дисертаційних магістерських кваліфікаційних робіт. А також передує вивченню дисципліни «Методологія інженерії програмного забезпечення»

3. Зміст навчальної дисципліни

Курсова робота виконується за індивідуальним завданням і готується до захисту в завершальний період теоретичного навчання. Курсова робота повинна бути підготовлена до захисту в термін, обумовлений у завданні та погодженому з викладачем. До захисту курсової роботи представляється пояснювальна записка у складі:

- титульна сторінка;
- завдання на курсову роботу;
- календарний план-графік виконання роботи;
- анотація українською мовою;
- зміст;
- перелік скорочень, умовних позначень;
- вступ;
- основна частина;
- висновки;
- список використаних джерел;
- додатки.

Основна частина поділяється на розділи, у яких висвітлюються наступні моменти:

- огляд та аналіз існуючих рішень;
- формулювання вимог до нового продукту (системи);
- обґрунтування та опис архітектури системи;
- вибір технологій та фреймворків для реалізації продукту (системи);
- опис конструювання, розгортання та функціонування системи;
- тестування продукту.

Для опису (моделювання) побудови та функціонування системи та її складових використовувати схеми згідно Державним стандартам України щодо оформлення програмної документації, стандартам ISO а також діаграми UML.

4. Навчальні ресурси та матеріали

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. Роберт Мартін. Чистий Agile. Переклад укр. Видавництво Фабула, 2021, 224 с. ISBN: 978-617-09-6760-2
3. Software Engineering Body of Knowledge (SWEBOK)
<https://www.computer.org/education/bodies-of-knowledge/software-engineering>
4. Стандарти:
ISO/IEC/IEEE 12207 Systems and software engineering — Software life cycle processes
ISO/IEC/IEEE 29119-4:2021 Software and systems engineering — Software testing
<https://www.iso.org/standards.html>
ДСТУ ISO/IEC/IEEE 15288:2016 Інженерія систем і програмного забезпечення. Процеси життєвого циклу систем
https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71827
ДСТУ ISO/IEC/IEEE 16326:2015 Розроблення систем та програмного забезпечення. Процеси життєвого циклу. Керування проектами
https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=67052
ДСТУ 3008-2015 Державний стандарт України. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання
ДСТУ ISO 5457:2006 (ISO 5457:1999, IDT). Державний стандарт України (ISO). Документація технічна на виробі. Кресленики. Розміри та формати
5. Martin Fowler et al. Patterns of Enterprise Application Architecture. Addison Wesley, 2002, 560 p. ISBN 0-321-12742-0
6. Steve McConnell. More Effective Agile: A Roadmap for Software Leaders. Construx Press, 2019, 380 p. ISBN: 978-1-7335-1821-5
7. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, Ed. 2000, 368 p.
8. 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
9. Steve McConnell. Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft Press, 2006, 308 p. ISBN: 978-0-7356-0535-0
10. Steve McConnell. Professional Software Development: Shorter Schedules, Higher Quality Products, More Successful Projects, Enhanced Careers. Addison Wesley, 2003, 272 p. ISBN: 9780321193674
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

Інформаційні ресурси

1. The Unified Modeling Language [Online] Available from: <https://www.uml-diagrams.org/>
2. Катерина Шевченко. Створення асинхронної SOA для застосунків: потенційні проблеми та рішення. 11 жовтня 2023 р. [Online] Available from: <https://journal.gen.tech/post/stvorenniya-asinhronnoyi-soa-dlya-zastosunkiv-potencijni-problemi-ta-rishennya>

Приклади коду:

- Actor model: <https://github.com/HowProgrammingWorks/Actor>
- Pattern Actor: <https://github.com/HowProgrammingWorks/Actor>
- Domain-driven design: <https://github.com/HowProgrammingWorks/DDD>
- Blockchain: <https://github.com/HowProgrammingWorks/Blockchain>
- Conflict-free Replicated Data Types: <https://github.com/HowProgrammingWorks/CRDT>
- Peer-to-Peer with WebRTC: <https://github.com/HowProgrammingWorks/WebRTC>
- Paradigms Comparison: <https://github.com/HowProgrammingWorks/Paradigms>
- Metaprogramming: <https://github.com/HowProgrammingWorks/Metaprogramming>
- Modularity: <https://github.com/HowProgrammingWorks/Modularity>
- Dependency Injection: <https://github.com/HowProgrammingWorks/DependencyInjection>
- Unittesting: <https://github.com/HowProgrammingWorks/Unittesting>
- Abstractions: <https://github.com/HowProgrammingWorks/Abstractions>
- Project: <https://github.com/HowProgrammingWorks/Project>
- Abstraction Layers: <https://github.com/HowProgrammingWorks/AbstractionLayers>
- Sandboxes: <https://github.com/HowProgrammingWorks/Sandboxes>
- Inversion Of Control: <https://github.com/HowProgrammingWorks/InversionOfControl>
- Dependency Injection: <https://github.com/HowProgrammingWorks/DependencyInjection>
- Inter Process Communication:
<https://github.com/HowProgrammingWorks/InterProcessCommunication>
- Data access layer: <https://github.com/HowProgrammingWorks/Transaction>
- API: <https://github.com/HowProgrammingWorks/API>
- CQS: <https://github.com/HowProgrammingWorks/CQS>
- CQRS: <https://github.com/HowProgrammingWorks/CQRS>
- EventSourcing: <https://github.com/HowProgrammingWorks/EventSourcing>
- Pattern Command: <https://github.com/HowProgrammingWorks/Command>
- SAGA Pattern: <https://github.com/HowProgrammingWorks/Saga>
- Referential Transparency: <https://github.com/HowProgrammingWorks/ReferentialTransparency>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компоненту)

На виконання курсової роботи заплановано загальною 30 годин самостійної роботи впродовж 18 тижнів навчального семестру.

Календарний план виконання курсової роботи

Тиждень	Етапи виконання
1	Отримання теми курсової роботи, опису завдань, узгодження з викладачем
2	Системний аналіз завдання. Пошук, огляд та аналіз літератури, стандартів, інформаційних джерел.

3	Загальне формулювання вимог. Моделювання та аналіз функціональних вимог. Діаграми прецедентів. Формулювання вимог до інтерфейсу користувача та (або) фасаду системи. Розробка алгоритмів верхнього рівня. Блок-схеми алгоритмів.
4	Проектування архітектури системи. Розробка, моделювання та аналіз вимог до архітектури. Опис компонентів і модулів та їхньої взаємодії. Діаграми послідовностей.
5	Детальне проектування системи. Вибір та обґрунтування використання патернів проектування. Об'єктно-орієнтований аналіз та моделювання. Діаграми класів. Схеми баз даних. Алгоритми деталізованої поведінки модулів та фрагментів системи.
6-8	Конструювання програмного застосунку (системи). Вибір та обґрунтування використання технологій розробки, фреймворків, сторонніх компонентів та бібліотек. Програмування, створення програмного коду.
9-10	Планування тестування. Визначення обсягу та етапів тестування. Визначення середовища та потрібні ресурси для тестування. Розробка тест-кейсів. Виконання функціонального, модульного, навантажувального тестування.
11	Завершення циклу тестування. Аналіз результатів тестування.
12	Виправлення помилок (за необхідності). Рефакторинг (за необхідності). Повторний цикл тестування (за необхідності)
13	Оформлення пояснювальної записки
14	Попередня перевірка результатів виконання курсової роботи
15	Захист курсової роботи

Основні положення щодо виконання роботи:

- Тема курсової роботи може бути узгодженою з темою майбутньої кваліфікаційної роботи.
- Етапи курсової роботи повинні бути виконані згідно встановленого календарного графіку робіт. Передбачена також можливість повторення – ітеративність деяких етапів
- Розроблений програмний застосунок (система) повинен бути протестований. Результати тестування програмного додатку наводяться у тексті основної частини пояснювальної записки курсової роботи.
- Оформлення курсової роботи має відповідати вимогам до звітів про НДР (ДСТУ3008-2015). В графічних ілюстраціях можуть також застосовуватися діаграми UML.
- Увесь ілюстративний матеріал у курсовій роботі повинен бути виконаний за допомогою комп'ютерних засобів. Зміст ілюстративного матеріалу має з достатньою повнотою відображати основні положення, які виносяться на захист.
- У випадку виявлення факту академічної недоброчесності та плагіату курсова робота повертається на докорінну переробку з можливою зміною теми.

6. Самостійна робота здобувача вищої освіти

Самостійна робота здобувача вищої освіти під час виконання курсової роботи охоплює отримання та усвідомлення теми й завдання, пошук, опрацювання й аналіз наукових і інформаційних джерел, а також розробку програмного застосунку (системи) з подальшим документуванням результатів. Важливою складовою є оформлення пояснювальної записки відповідно до встановлених вимог і підготовка до захисту курсової роботи, що передбачає узагальнення отриманих результатів та аргументоване представлення виконаної роботи.

Загальний обсяг самостійної роботи 30 годин.

7. Політика навчальної дисципліни (освітнього компонента)

Як викладач, так і студент зобов'язані дотримуватись Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>),

Можливість дистанційного режиму. 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://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах Zoom, Google Meet, BigBlueButton. Посилання на конференцію видається студентам на початку семестру. З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський». Результати оцінювання висвітлюють у АС «Електронний кампус» у відповідних відомостях поточного, календарного та підсумкового контролю.

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 плагиату в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/47>). У разі виявлення дублювання робіт, плагиату 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.

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

Семестровий контроль – залік, який проводиться у вигляді усного публічного захисту курсової роботи (КР) перед комісією з проведення семестрового контролю.

Поточний контроль – проводиться відповідно завершенню етапів календарного графіка виконання КР.

Календарний контроль проводиться два рази на семестр відповідно затверджених деканатом термінів його проходження. В ході календарного контролю перевіряється дотримання виконання етапів КР відповідно календарному графіку (технічному завданню).

Підсумкова оцінка ($r_{кр}$) за курсову роботу складається з суми двох складових:

1. Стартової складової ($r_{старт}$). Максимальна кількість балів для неї 50 балів. Стартова складова для оцінки якості створеного програмного застосунку та вмісту та оформлення пояснювальної записки.

2. Складова захисту ($r_{захисту}$). Максимальна кількість балів для неї 50 балів. Складова захисту для

оцінки доповіді, рівня володіння матеріалом, обґрунтованості прийнятих рішень, вміння захищати свою думку, відповідей на запитання членів комісії

Критерії оцінювання стартової складової:

- відповідність програмного застосунку (системи) функціональним вимогам – 5 балів;
- повнота тестування, відсутність випадків некоректної роботи (збоїв) застосунку (системи) – 5 балів;
- виконання вимог щодо розгортання програмного застосунку (системи) – 5 балів;
- повнота опису (моделювання) архітектури програмного застосунку (системи) – 5 балів;
- повнота та коректність опису функціонування програмного застосунку (системи) – 5 балів;
- якість оформлення текстової частини та графічних матеріалів пояснювальної записки – 5 балів;
- дотримання вимог стандартів та нормативних документів – 5 балів;
- якісне конструювання, досконалість програмного коду, грамотне використання та реалізація патернів, наочне коментування – 5 балів;
- повнота огляду та аналізу існуючих рішень, аналогів, наявність та правильне оформлення посилань на літературні джерела та інформаційні ресурси – 5 балів;
- повнота обґрунтування проєктного рішення та використаних технологій – 5 балів

Критерії оцінювання складової захисту:

- повнота та наочність презентації роботи – 10 балів;
- демонстрація повноти та успішності відпрацювання програмним застосунком (системою) функціоналу згідно вимог – 10 балів.
- правильність та повнота відповідей на запитання, рівень опанування теретичних основ та нвбуття практичних навичок та виїнь щодо використання сучасних засобів створення програмного забезпечення – 30 балів

Таким чином, підсумкова оцінка за курсову роботу

$$r_{кр} = r_{старт} + r_{захисту}$$

Умови допуску до захисту. До захисту не допускаються курсові роботи, у яких відсутнє відповідне до завдання програмне забезпечення або немає пояснювальної записки. Також до захисту не допускаються курсові роботи, у яких стартова складова виконана менше ніж на 60%, тобто $r_{старт}$ менше 30 балів.

Таблиця переведення підсумкової оцінки ($r_{кр}$) в університетську шкалу оцінок

Кількість балів	Університетська шкала оцінок
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску до захисту	Не допущено

Робочу програму навчальної дисципліни (силабус)

Складено: доцент кафедри ОТ, к.т.н., доц. Волокита Артем Миколайович,
старший викладач кафедри ОТ Шемсєдинов Тимур Гафарович.

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

Погоджено методичною комісією ФІОТ (протокол №11 від 27.06.2025).