



Методологія інженерії програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти

Другий (магістерський)

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>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредитів /120 год. Лекцій 32 годин, Лабораторні роботи - 14 годин, Самостійна робота студента - 76 годин</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року за адресою https://schedule.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Керівник (забезпечує курс): Новотарський Михайло Анатолійович, професор кафедри обчислювальної техніки, д.т.н., професор Викладач: старший викладач кафедри обчислювальної техніки <i>Шемсединов Т.Г. timur.shemsedinov@gmail.com</i>
Розміщення курсу	https://classroom.google.com/c/ODM4Mzg5MTMxNTY4?cjc=f3ma7s2g

Програма навчальної дисципліни

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

Метою навчальної дисципліни є формування у студентів здатностей (компетентностей) розробки структури та архітектури програмних систем що забезпечує їх зручне розширення, модифікацію та системну інтеграцію. За результатами вивчення дисципліни студент має бути здатним вирішувати професійні завдання та володіти такими компетентностями:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01);
- Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК01);
- Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів (ФК03);

- Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення (ФК05);
- Здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення (ФК06);
- Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах (ФК07);
- Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення (ФК08);
- Здатність забезпечувати якість програмного забезпечення (ФК09)

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

- Знати і застосовувати сучасні професійні стандарти та нормативно-правові документи з інженерії програмного забезпечення (ПРН01)
- Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу (ПРН02)
- Будувати і досліджувати моделі інформаційних процесів у прикладній області (ПРН03)
- Виявляти інформаційні потреби і класифікувати дані для проєктування програмного забезпечення (ПРН04)
- Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення (ПРН05)
- Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів (ПРН06)
- Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення (ПРН07)
- Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника (ПРН08)
- обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення (ПРН09);
- Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проєктування програмного забезпечення (ПРН10)
- Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення (ПРН11)
- Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу (ПРН13)
- Прогнозувати розвиток програмних систем та інформаційних технологій (ПРН14)
- Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника (ПРН15)
- Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення (ПРН16)

- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела (ПРН17);
- Планувати і виконувати наукові дослідження в сфері інженерії програмного забезпечення, обирати методики та інструменти, аналізувати результати, обґрунтовувати висновки (ПРН20)

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

Пререквізити: Розроблення проблемно-орієнтованих та сервісно-орієнтованих систем.

Постреквізити: Інтеграція застосувань на основі мікросервісного підходу. Також знання і уміння можуть бути використані при виконання магістерської дисертації.

3. Зміст навчальної дисципліни

Розділ 1. Проєктування архітектури програмних систем

Тема 1. Архітектурний підхід до побудови систем

Тема 2. Архітектура для Web додатків.

Тема 3. Інтеграція систем та топологія інформаційних систем.

Тема 4. Кластеризація, паралельні та розподілені системи

Тема 5. Системи базовані на подіях та повідомленнях: MQ, Pub/Sub, Pull

Тема 6. Паттерни Command, QueryObject, CQS, CQRS, EventSourcing.

Тема 7. Методологія Feature-Sliced Design

Розділ 2. Конструювання та тестування програмного забезпечення

Тема 8. Модульність ПЗ, модульне тестування

Тема 9. Модель акторів та патерни Actor, Reactor, Proactor.

Тема 10. Моделювання та захист даних.

Тема 11. Використання вбудованих мов

Тема 12. Комунікаційні стилі

Тема 13. Організація доступу до даних.

Тема 14. Асинхронне програмування, паралельне та конкурентне програмування

Тема 15. Принцип “Imperative shell, Functional core”

Тема 16. Мультипарадигмене програмування та його вплив на архітектуру інформаційних систем

Тема 17. Метапрограмування: прийоми та техніки

4. Навчальні матеріали та ресурси

Базова література:

1. Матеріали дистанційного курсу – розміщено за адресою:
<https://classroom.google.com/c/ODM4Mzg5MTMxNTY4?cjc=f3ma7s2g>
2. Шемсєдинов Т.Г. Software Structure & Architecture. [Електронний ресурс] розміщено за адресою: <https://github.com/HowProgrammingWorks/Index/blob/master/Courses/Architecture-2025.md>
3. 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/>

Додаткова література:

1. Роберт Мартін. Чистий Agile. Переклад укр. Видавництво Фабула, 2021, 224 с. ISBN: 978-617-09-6760-2
2. Martin Fowler et al. Patterns of Enterprise Application Architecture. Addison Wesley, 2002, 560 p. ISBN 0-321-12742-0
3. М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д.Робертс. Рефакторинг. Поліпшення існуючого коду. К: Діалектика, 2003. – 448 с.
4. Steve McConnell. More Effective Agile: A Roadmap for Software Leaders. Construx Press, 2019, 380 p. ISBN: 978-1-7335-1821-5
5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, Ed. 2000, 368 p.
6. 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
7. Steve McConnell. Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft Press, 2006, 308 p. ISBN: 978-0-7356-0535-0
8. Steve McConnell. Professional Software Development: Shorter Schedules, Higher Quality Products, More Successful Projects, Enhanced Careers. Addison Wesley, 2003, 272 p. ISBN:9780321193674
9. Robert C. Martin. Clean Agile: Back to Basics. Longman, 2019, 224p. ISBN: 9786170967602
10. Robert C. Martin and Micah Martin. Agile Principles, Patterns, and Practices in C#. Prentice Hall, 2006, 768p. ISBN:978-0-13-185725-4
11. 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>
12. 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>
13. Kent Beck, Cynthia Andres. Extreme Programming Explained. 2nd Edition. Addison Wesley, 2004, 224 p. ISBN: 978-0321278654

Інформаційні ресурси

SWEBOK

- Software Engineering Body of Knowledge (SWEBOK)
<https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- Maxym Zosym. SWEBOK v3. 04 May 2023.
<https://www.maxzosim.com/swbok-v3/>

PMBOK

- Стандарт з управління проєктами та настанова до зводу знань з управління проєктами (Настанова РМВОК Сьоме видання). Переклад укр. Обмеження: лише для особистого використання. Джерело перекладу: A Guide to the Project Management Body of Knowledge:

Офіційні 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
ISO/IEC/IEEE 90003:2018 Software engineering — Guidelines for the application of ISO 9001:2015 to computer software
<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

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекції

№	Назва теми лекції Основні питання	Кількість ауд. годин
1	Тема 1. Архітектурний підхід до побудови систем Огляд матеріалів курсу, підходів, парадигм та засобів розробки. Тема 2. Архітектура для Web додатків на базі Local-First DDD для використання на Frontend та Backend. Шари у DDD архітектурі, Layered (onion), DDD, Clean architecture, зв'язаність та зв'язність коду, принципи GRASP та SOLID у сервісах.	2
2	Тема 3. Інтеграція систем та топологія інформаційних систем API, шини даних, брокери, MQ (черги повідомлень). Гексагональна архітектура, архітектура портів та адаптерів.	2
3	Тема 4. Кластеризація, паралельні та розподілені системи CAP, ACID, BASE, блокування, CQRS. Розподілені системи та CAP-теорема, узгодженість, доступність та розподіленість.	2
4	Тема 5. Системи базовані на подіях та повідомленнях MQ, Pub/Sub, Pull	2
5	Тема 6. Паттерни Command, QueryObject, CQS, CQRS, EventSourcing.	2
6	Тема 7. Методологія Feature-Sliced Design	2
7	Тема 8. Модульність ПЗ, модульне тестування Технологія DI (впровадження залежностей)	2
8	Тема 9. Модель акторів та патерни Actor, Reactor, Proactor	2
9	Тема 10. Моделювання та захист даних DTOs, моделювання даних, race conditions та захист даних у великих системах, ізоляція шарів та підсистем	2

10	Тема 11. Використання вбудованих мов Domain Specific Languages (DSL), спеціальних синтаксисів (ECMAScript, JSON, MD, LISP) та AST для опису домену.	2
11	Тема 12. Організація доступу до даних. Патерни доступу до даних (Data access patterns).	2
12	Тема 13. Комунікаційні стилі data, call, event, log sync (синхронізація логів), p2p (пірінові чи однорангові системи), blockchain. Безконфліктні репліковані типи даних (CRDT).	2
13	Тема 14. Асинхронне програмування, паралельне та конкурентне програмування	2
14	Тема 15. Принцип “Imperative shell, Functional core”	2
15	Тема 16. Мультипарадигмене програмування та його вплив на архітектуру інформаційних систем	2
	Тема 17. Метaprogramування Прийоми та техніки, інтроспекція, рефлексія, скафолдінг. Динамічна інтерпретація метамоделі.	2
16	Модульна контрольна робота	2
	Всього:	32

Лабораторні роботи

№	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд.годин
1	Аналіз вимог та розробка архітектури Web-системи за методологією Local-First. Моделювання архітектури синхронізації даних на базі подій	2
2	Конструювання Backend вузла розподіленої системи для синхронізації даних через CRDT, blockchain та протоколів WebRTC, WebSocket, та HTTP/3	4
3	Модульне тестування Back-End, та системних компонентів (DSL, MQ черг повідомлень, DAL шару доступу до даних)	2
4	Конструювання Frontend вузла розподіленої системи для синхронізації даних через CRDT, blockchain та протоколів WebRTC, WebSocket, та HTTP/3	2
5	Валідація блоків Blockchain, виконання бізнес-логіки у smart-контрактах, використання State-based CRDT: G-Counter, PN-Counter, Δ -G-Counter, Δ -LWW-Set, Δ -MV-Register, Δ -RGA, Δ -LWW-Register та інших.	2
6	Організація доступу до даних DAL, Розгортання системи, конфігурування, тестування під навантаженням, тестування вирішення конфліктів при зникненні мережі та відновленні синхронізації.	2
	Разом:	14

Розрахунково-графічна робота

Розрахунково-графічна робота (РГР) планується у вигляді проєкту створення конкретного програмного застосунка і виконується в ході самостійної роботи. Для кожного здобувача індивідуальне завдання.

Здобувачі можуть запропонувати власні теми для РГР відповідно тематичному змісту даної дисципліни, наведеного у цьому силабусі.. Тема та завдання на РГР узгоджується не пізніше як за два місяця до кінця семестру. Здавання РГР у формі доповіді-презентації та співбесіди з викладачем.

6. Самостійна робота студента/аспіранта

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 – 0.5 годин на кожну лекцію	8
2	Виконання лабораторних робіт з вивченням теорії та реалізацією наведеної теми у програмному коді та підготовка до захисту. У середньому 6 годин на роботу	36
3	Розрахунково-графічна робота	20
4	Підготовка до МКР	4
5	Підготовка до заліку	6
	Всього:	74

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://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-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 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. 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 коміти, чітко та зрозуміло описувати коміти та пул-реквести, додавати теги для лінування issues, PR, та комітів 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;

Лабораторні роботи та РГР 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.

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

Види контролю з навчальної дисципліни «Методологія інженерії програмного забезпечення» 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 лабораторних робіт.

Критерії оцінювання виконання завдань лабораторних робіт:

- повна відповідь, не менше 90% потрібної інформації, проєкт на Github містить також текст опису проєкта – 9-10 балів;
- достатньо повна відповідь, не менше 75% потрібної інформації, незначні неточності, опис проєкта на Github неповний – 7-8 балів;
- неповна відповідь, не менше 60% потрібної інформації, значні неточності у відповіді по суті питання або не наданий опис проєкта на Github – 6 балів;
- у відповіді менше 60% потрібної інформації, суттєві неточності, помилки, некоректна робота програмного застосунку – 0-5 балів.

Модульна контрольна робота у вигляді 2 теоретичних запитань по тематиці дисципліни. За відповідь на кожне запитання максимальна оцінка 5 балів.

Критерії оцінювання відповідей на запитання МКР:

- 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 бали.

Розрахунково-графічна робота.

Критерії оцінювання виконання РГР:

- змістовна доповідь-презентація, не менше 90% потрібної інформації, проєкт на Github містить повний текст опису проєкта – 27-30 балів;

- неповна доповідь-презентація, не менше 75% потрібної інформації, незначні неточності, опис проєкта на Github неповний, тестування не у повному обсязі – 21-26 балів;
- неповна доповідь-презентація, не менше 60% потрібної інформації, значні неточності у відповідях по виконанню роботи, опис проєкта на Github неповний, некоректне тестування – 18-20 балів;
- недостатня доповідь-презентація, менше 60% потрібної інформації, суттєві неточності, помилки, некоректна робота програмного застосунку – 0-17 балів.

Індивідуальний семестровий рейтинг студента є сумою балів, отриманих студентом впродовж семестру відповідно таблиці.

Оцінювання окремих видів навчальної роботи студента

Вид навчальної роботи	Максимальна кількість балів
Лабораторна робота № 1	10
Лабораторна робота № 2	10
Лабораторна робота № 3	10
Лабораторна робота № 4	10
Лабораторна робота № 5	10
Лабораторна робота № 6	10
Модульна контрольна робота	10
Розрахунково-графічна робота	30
Всього:	100

Семестровий контроль

Умовою виконання курсу є зарахування усіх лабораторних робіт, РГР та МКР.

Ті здобувачі, які виконали цю умову та їх індивідуальний семестровий рейтинг (R_D) не менше 60 балів, для них оцінка R_D буде підсумковим семестровим рейтингом. Але, ці здобувачі за їхнім бажанням можуть також проходити залік з метою підвищення підсумкового рейтингу.

Здобувачі, які виконали умову виконання курсу, але їхній семестровий рейтинг $40 \leq R_D < 60$, повинні проходити залік.

Здобувачі, які не виконали умову виконання курсу або їхній рейтинг $R_D < 40$, до заліку не допускаються.

Залік

Відбувається згідно умов, наведених вище.

Залік проводиться у формі співбесіди здобувача з викладачем. Викладач може задавати запитання відповідно тематичному змісту дисципліни, зокрема, зокрема по тим темам, які здобувачем були неповністю опрацьовані впродовж семестру. Максимальна залікова оцінка 20 балів.

Критерії оцінювання заліку:

- повна відповідь на усі запитання, 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 відповідях менше 60% потрібної інформації, суттєві неточності – 0-11 балів. Така оцінка не зараховується.

У разі проходження заліку підсумкова семестрова рейтингова оцінка здобувача буде сумою балів індивідуального семестрового рейтингу і залікової оцінки.

Підсумкова семестрова рейтингова оцінка перераховується в оцінки за національною шкалою та шкалою ECTS

Відповідність рейтингових балів оцінкам за університетською шкалою

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Є незараховані лабораторні роботи, РГР та/або МКР	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено: професор кафедри обчислювальної техніки, д.т.н., проф. Новотарський М.А.
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.)