



ФОРМАЛЬНІ МЕТОДИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Третій (PhD)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>5 кредитів (150 год., з них 14 год. лекції, 28 год. комп'ютерний практикум, 108 год. самостійна робота)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен / МКР</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>д.т.н., професор, Стеценко Інна Вячеславівна, i.stetsenko@kpi.ua, inna.stetsenko-fiot@iit.kpi.ua</i> Практичні / Семінарські: <i>немає за навчальним планом</i> Лабораторні: <i>Стеценко Інна Вячеславівна</i>
Розміщення курсу	https://drive.google.com/drive/folders/1lclgC5cab2Wl9SPbg9eOhgV0dFoXjt

Програма навчальної дисципліни

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

Вивчення дисципліни спрямовано на оволодіння методами розробки моделей програмних систем та їх дослідження з метою аналізу швидкодії, надійності, безпечності обчислювальних процесів. Розглядаються формалізми для побудови моделей, такі як алгебри процесів та високорівневі мережі Петрі, методи моделювання паралельних та розподілених обчислень, методи дослідження взаємодії процесів у програмних системах, а також експериментальне визначення ефективності програмних систем на основі моделей. Приклади моделей, які наводяться, покривають моделі паралельних алгоритмів, розподілених інформаційних систем, передачі даних у телекомунікаційних системах, розповсюдження шкідливого програмного забезпечення та інші. Дисципліна розвиває та удосконалює уміння будувати моделі програмних систем для наукового дослідження та застосовувати їх для отримання наукових результатів, надає необхідні математичні знання з моделювання взаємодії процесів. Завдання комп'ютерного практикуму спираються на навички розробки алгоритмів імітації та проведення експериментального дослідження моделі.

Предмет навчальної дисципліни – методи проектування, верифікації та дослідження ефективності програмних систем на основі моделей.

Метою дисципліни є вивчення студентами принципів та способів розробки, реалізації та дослідження формальних моделей програмних систем, надбання навичок використання

формальних методів дослідження програмних систем з метою підвищення їх швидкодії, надійності, а також безпечності.

За результатами вивчення дисципліни здобувач вищої освіти повинен **набути досвід** формалізації взаємодії процесів у програмних системах та використання моделей програмних систем для наукового дослідження.

Студент після засвоєння навчальної дисципліни повинен **знати**:

- методи та способи формалізації моделей програмних систем,
- методи моделювання обчислювальних процесів мережами Петрі (базовими, стохастичними, високорівневими),
- методи побудови Петрі-об'єктних моделей програмних систем,
- моделі паралельних та розподілених обчислень та їх дослідження,
- методи дослідження ефективності програмних систем на основі формальних методів,
- методи алгебри процесів (process algebras) та їх використання для дослідження взаємодії процесів,
- методи темпоральної логіки та їх використання для верифікації програм,
- сучасні тенденції розвитку формальних методів програмної інженерії

Студент повинен **вміти**:

- розробляти моделі програмних систем засобами базових та стохастичних мереж Петрі,
- розуміти моделі систем, побудованих з використанням розфарбованих мереж Петрі,
- розробляти Петрі-об'єктні моделі програмних систем,
- складати формалізовані моделі програмних систем з використанням алгебри процесів,
- застосовувати методи алгебри процесів для дослідження взаємодії обчислювальних процесів,
- оцінювати точність та складність побудованих моделей,
- оцінювати ефективність паралельних та розподілених обчислень з використанням формальних моделей,
- досліджувати ефективність програмних систем з використанням формальних методів.

Згідно з вимогами освітньої програми вивчення дисципліни спрямоване на оволодіння студентом таких **компетентностей**:

ФК01 Здатність виконувати оригінальні дослідження, досягати наукових результатів, які створюють нові знання у інженерії програмного забезпечення та дотичних до неї міждисциплінарних напрямках і можуть бути опубліковані у провідних наукових виданнях з інформаційних технологій та суміжних галузей.

ФК05 Здатність до розроблення нових та вдосконалення існуючих моделей, методів, засобів, процесів у сфері програмної інженерії, які забезпечують розвиток або надають нові можливості технологіям розроблення та використання програмного забезпечення.

ФК06 Здатність застосовувати формальні методи проектування, розроблення та дослідження програмних систем та технологій у наукових дослідженнях.

Програмні результати вивчення дисципліни забезпечують такі програмні результати освітньої програми:

ПРН01 Мати передові концептуальні та методологічні знання з інженерії програмного забезпечення і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.

ПРН03 Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у інженерії програмного забезпечення та дотичних міждисциплінарних напрямках.

ПРН06 Розуміти теоретичні засади, що лежать в основі методів досліджень інформаційних систем та програмного забезпечення, методології проведення досліджень та обчислювальних експериментів.

ПРН08 Уміти створювати програмне середовище комп'ютерного моделювання інформаційних систем.

ПРН09 Знати методології комп'ютерного моделювання складних систем.

ПРН10 Знати методології візуального моделювання складних систем.

ПРН11 Знати принципи побудови сценарних моделей та верифікації сценаріїв аналізу інформації.

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

Вивчення дисципліни спирається на знання, отримані здобувачами при вивченні дисципліни «Організація науково-інноваційної діяльності», знання та навички, набуті студентом при вивченні дисципліни «Методи реінжинірингу програмного забезпечення». Знання та навички, отримані при вивченні дисципліни, спрямовані на використання у науковому дослідженні доктора філософії.

3. Зміст навчальної дисципліни

Тема 1.1. Огляд формальних методів інженерії програмного забезпечення.

Тема 1.2. Формалізм мереж Петрі. Класична мережа Петрі.

Тема 1.3. Стохастична мережа Петрі.

Тема 1.4. Технологія Петрі-об'єктного моделювання: конструювання моделі, алгоритм імітації, інтерпретація результатів моделювання.

Тема 1.5. Моделювання паралельних обчислень з використанням класичних та стохастичних мереж Петрі, Петрі-об'єктних моделей.

Тема 1.6. Програмні засоби для моделювання програмного забезпечення мережами Петрі: Tina, CPN IDE, Renew, PetriObjModelPaint.

Тема 1.7. Інші формальні методи для проектування обчислень в програмній інженерії: алгебри процесів, формалізм Мілнера, π -числення.

4. Навчальні матеріали та ресурси

Базова література

1. Нестеренко Б.Б., Новотарський М.А. Формальні засоби моделювання паралельних процесів та систем. Праці Інституту математики НАН України, т.90. Київ: Ін-т математики НАН України, 2012. 334 с.
2. Hoare C.A.R., Jifeng H. *Unifying Theories of Programming*. New York: Prentice-Hall, 1998. — 311 p.
3. Jensen K. *Coloured Petri Nets: Modeling and Validation of Concurrent Systems* / K.Jensen, L.Kristensen - Springer-Verlag Berlin Heidelberg, 2000 – 383p.
4. Stetsenko I.V., Dyfuchyna O. (2020) Thread Pool Parameters Tuning Using Simulation. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) *Advances in Computer Science for Engineering and Education II. ICCSEE 2019. Advances in Intelligent Systems and Computing*, vol 938, 78-89. Springer, Cham. 2020.
5. Stetsenko I.V. *Simulation of Multithreaded Algorithms Using Petri-Object Models*. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) *Advances in Computer Science for Engineering and Education. ICCSEE 2018. Advances in Intelligent Systems and Computing*, vol 754, 391-401. Springer, Cham. 2020.
6. Юрчишин В.М., Піх В.Я Формальні методи програмної інженерії: конспект лекцій. – Івано-Франківськ: ІФТУНГ, 2016. – 154 с.
7. Hou Y., Barkaoui K. *Deadlock analysis and control based on Petri nets: A siphon approach review*. *Advances in Mechanical Engineering* 2017, Vol. 9(5) 1–30.
8. Nicola R. A gentle introduction to Process Algebras <https://pdfs.semanticscholar.org/12d9/eae1638729aeb237b5be445ee91ecdd3c5d7.pdf>

Додаткова література

1. Pierce B.C., Amorim A.A., Casinghino, C. Gaboardi M., Greenberg M., Hrițcu C., Sjöberg V., Yorgey B. Software foundations. Vol. 2 (2019) <https://softwarefoundations.cis.upenn.edu/plf-current/index.html>
2. Peters K. Comparing Process Calculi Using Encodings In: J.A. Pérez and J. Rot (Eds.): Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS 2019). EPTCS 300, 2019, pp. 19–38.
3. Concurrent object-oriented programming and Petri nets: advances in Petri nets / Gul A. Agha. (ed.). - Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 2001.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
1.	Лекція 1 Огляд формальних методів інженерії програмного забезпечення.	– Переваги та недоліки застосування формальних методів аналізу програмного забезпечення на етапах його проектування, розробки та експлуатації. – Процес формалізації дискретно-подійних процесів. – Методи аналізу процесів на основі протоколу подій. – Протоколи подій та їх використання для побудови моделей складних програмних систем. – Сучасні тенденції в розробці нових засобів формалізації складних систем: ієрархічні моделі, багаторівневі моделі.
2.	Комп'ютерний практикум 1. UML-опис програмного забезпечення	Завдання: – Визначити переваги та недоліки опису програмного забезпечення засобами UML. – Виконати аналіз властивостей програмного забезпечення, що розробляється у дисертаційному дослідженні, на основі його опису поведінковими UMLдіаграмами.
3.	Лекція 2 Класичні мережі Петрі.	– Математичне визначення класичної мережі Петрі. – Основні положення теорії класичних мереж Петрі. – Властивості мереж Петрі та їх дослідження матричним способом. Інваріанти мережі Петрі та алгоритми їх пошуку. – Дерево досяжності та алгоритми його побудови.
4.	Комп'ютерний практикум 2. Математичне моделювання процесів передачі даних класичною мережею Петрі	Завдання: – Розробити 1) модель фрагменту програмного коду, що розробляється за темою дисертаційного дослідження, у вигляді класичної мережі Петрі; 2) модель філософів, що обідають. – Скласти математичний опис розроблених моделей. – Дослідити властивості розроблених мереж Петрі: обмеженість, досяжність, консервативність, циклічність.
5.	Комп'ютерний практикум 3. Імітаційне	Завдання:

	моделювання процесів передачі даних класичною мережею Петрі	– У програмному забезпеченні з моделювання мереж Петрі Тіпа розробити імітаційну модель передачі даних телекомунікаційної системи. – Використовуючи імітаційну модель дослідити можливість виникнення тупикової ситуації в системі передачі даних. – Визначити переваги та недоліки програмного забезпечення Тіпа для дослідження властивостей мережі Петрі.
6.	Лекція 3 Стохастичні мережі Петрі.	– Часові затримки та способи їх завдання. Конфліктні переходи та алгоритми вирішення конфліктів. – Математичні рівняння станів стохастичної мережі Петрі. – Еквівалентні відношення між базовою, детермінованою та стохастичною мережами Петрі. Дослідження властивостей стохастичної мережі Петрі. – Розфарбована мережа Петрі. Програмне забезпечення CPN IDE. – Петрі-об'єктна модель: визначення, математичний опис, фундаментальні твердження, математичні рівняння станів. Програмне забезпечення PetriObjModelPaint.
7.	Комп'ютерний практикум 4. Математичне моделювання обчислювальних процесів стохастичною мережею Петрі	Завдання: – Розробити модель вебзастосунку стохастичною мережею Петрі. – Скласти математичні рівняння станів стохастичної мережі Петрі. – Дослідити досяжність маркірування з використанням математичних рівнянь.
8.	Комп'ютерний практикум 5. Імітаційне моделювання обчислювальних процесів стохастичною мережею Петрі	Завдання: – У програмному забезпеченні з моделювання стохастичних мереж Петрі CPN IDE або PetriObjModelPaint розробити імітаційну модель вебзастосунку. – Дослідити ефективність обчислювальних процесів веб застосування використовуючи імітаційну модель. – Порівняти результати, отримані на моделі, з результатами експериментального дослідження веб застосування
9.	Комп'ютерний практикум 6. Дослідження ефективності клієнт-серверного застосунку на його моделі	Завдання: – У програмному забезпеченні з моделювання стохастичних мереж Петрі CPN IDE або PetriObjModelPaint розробити модель клієнт-серверного застосунку з кількістю подій не менше 50. – Реалізувати модель з використанням програмного забезпечення CPN IDE, або PetriObjModelPaint, або Renew, в тому числі може бути власна розробка моделі, що складається з типових фрагментів мережі Петрі з часовими затримками. – Дослідити на моделі час отримання відповіді на запит при різних характеристиках апаратного забезпечення та різній інтенсивності запитів.

10.	Комп'ютерний практикум 7. Дослідження складності імітації стохастичної мережі Петрі	Завдання: - Для моделі клієнт-серверного застосування, розробленій у практикумі 6, дослідити зростання часу виконання алгоритму імітації при збільшенні інтервалу імітації та при збільшенні складності моделі (кількості подій в моделі). - Зробити висновки про застосування мереж Петрі для проєктування та розробки клієнт-серверних застосунків.
11.	Лекція 4 Моделювання паралельних обчислень з використанням класичних та стохастичних мереж Петрі, Петрі-об'єктних моделей.	- Основні механізми паралелізму та їх представлення мережами Петрі. - Методи пошуку дедлоків. - Верифікація та тестування паралельних алгоритмів з застосуванням моделей. - Аналіз ефективності. - Моделювання розподілених обчислень з використанням базових та стохастичних мереж Петрі, Петрі-об'єктних моделей. - Аналіз ефективності розподілених обчислень.
12.	Комп'ютерний практикум 8. Моделювання паралельного алгоритму класичною мережею Петрі	Завдання: - Розробити модель паралельного алгоритму класичною мережею Петрі. - Дослідити на моделі можливість виникнення дедлоків. - Дослідити взаємодію паралельних процесів на моделі.
13.	Комп'ютерний практикум 9. Імітаційне моделювання паралельної програми стохастичною мережею Петрі	Завдання: - Розробити модель паралельного алгоритму стохастичною мережею Петрі. - Дослідити на моделі ефективність паралельного алгоритму. - Порівняти результати, отримані на моделі, з результатами експериментального дослідження веб застосування
14.	Комп'ютерний практикум 10. Петрі-об'єктне моделювання паралельних обчислень	Завдання: - Розробити Петрі-об'єктну модель паралельних обчислень. - Дослідити на моделі ефективність застосування паралельних обчислень в залежності від кількості обчислювального ресурсу, параметрів алгоритму та складності алгоритму.
15.	Комп'ютерний практикум 11. Петрі-об'єктне моделювання розподілених обчислень	Завдання: - Розробити Петрі-об'єктну модель розподілених обчислень. - Дослідити на моделі ефективність застосування розподілених обчислень в залежності від архітектури розподіленої системи, кількості обчислювального ресурсу, параметрів обчислень та складності обчислень.
16.	Модульна контрольна робота	Контрольні тестові завдання
17.	Лекція 5. Формальні методи алгебри процесів.	- Формальний опис обчислювальних процесів. - Поняття про еквівалентність процесів.

		– Сучасні тенденції розвитку формальних методів алгебри процесів. – Теорія взаємодіючих послідовних процесів Гоара. Логіка Гоара.
18.	Комп'ютерний практикум 12. Моделювання взаємодії процесів з використанням логіки Гоара.	Завдання: – Розробити модель взаємодії процесів, що розглядались у попередніх практикумах, використовуючи логіку Гоара. – Виконати аналіз взаємодії процесів, використовуючи логіку Гоара.
19.	Лекція 6. Формалізм Мілнера для опису взаємодії систем. π -числення: основні поняття та правила виведення.	– Формалізація опису обміну повідомленнями в термінах π-числення. – Алгебра взаємодіючих процесів Бергстра і Клопа. – Порівняння трьох формалізмів алгебри процесів: Гоара, Мілнера, Бергстра і Клопа. – Формальний опис псевдопаралельних процесів з використанням алгебри процесів. – Парадигми програмування та формалізми опису обчислювальних процесів.
20.	Комп'ютерний практикум 13. Моделювання взаємодії процесів з використанням алгебри процесів Мілнера.	Завдання: – Розробити модель взаємодії процесів, що розглядались у попередніх практикумах, використовуючи алгебру процесів Мілнера. – Виконати аналіз взаємодії процесів, використовуючи алгебру процесів Мілнера
21.	Модульна контрольна робота	Контрольні тестові завдання

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді у гугл-класрумі <https://classroom.google.com/c/NTQ5OTEhOTE3OTE4?cjc=eauxhrrp>. Контент доступний із будь-якого місця в мережі Інтернет зареєстрованому на курс студенту. Навчальний курс складається з розділів, кожний з яких містить презентації лекцій, завдання та теоретичний матеріал для виконання комп'ютерного практикуму, URL-посилання на документацію з програмного забезпечення та навчальний матеріал для самостійної роботи відповідно до теми, яка вивчається.

Лекції по дисципліні проводяться викладачем дистанційно за розкладом. Відеозаписи лекцій після їх проведення розміщуються у гугл-класрумі. На лекціях проводяться експрес-опитування, які дають можливість лектору отримати інформацію про якість засвоєння матеріалу та, за необхідності, розглянути більш детально складний матеріал.

Комп'ютерні практикуми виконуються з використанням мови програмування Java, відкритого програмного забезпечення CPN IDE, Renew, а також програмного забезпечення PetriObjPaint, розміщеного в github.com/StetsenkoInna. Завдання кожного практикуму містить кілька підзавдань різної складності, які мають окреме оцінювання. Таке розбиття надає можливість більш об'єктивно оцінити рівень умінь здобувача вищої освіти і, водночас, адаптувати завдання до рівня знань та навичок студента. Для виконання кожного завдання комп'ютерного практикуму здобувачу надається не більше 4 тижнів. Здобувач на початку семестру отримує календарний план початку та завершення захисту завдань комп'ютерного практикуму. Альтернативою виконання завдання комп'ютерного практикуму може бути розробка моделі у відповідності до дисертаційного дослідження, яке проводить здобувач.

В умовах дистанційного навчання 2025-2026 н.р. усі види занять, у тому числі контрольні заходи, проводяться з використанням Zoom.

6. Самостійна робота студента/аспіранта

Матеріали для самостійного вивчення дисципліни розміщені викладачем у гугл-класі <https://classroom.google.com/c/NTQ5OTExOTE3OTE4?cjc=eauxhrrp>. Контент платформи доступний із будь-якого місця в мережі Інтернет. До самостійної роботи студента відноситься, в основному, виконання завдання комп'ютерного практикуму, робота з документацією та тьюторіалом програмного забезпечення, а також опрацювання лекційного та додаткового теоретичного матеріалу за наданими презентаціями лекцій, навчальним посібником та додатковою літературою. На самостійну роботу студент має витрати кількість годин, що дорівнює кількості годин, проведених ним на аудиторних заняттях.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Здобувач вищої освіти має вивчати дисципліну протягом семестру, дотримуючись календарного плану виконання завдань комп'ютерного практикуму, вивчення тем лекційного матеріалу та виконання модульних контрольних робіт. Усі завдання здобувач має виконувати **самостійно і вчасно**. Завдання вважається виконаним, якщо здобувач захистив завдання комп'ютерного практикуму у викладача та розмістив звіт з виконання у гугл-класі. Затримка у виконанні завдання більше ніж 4 тижні не допускається і можливість захистити завдання студентом втрачається назавжди. Такі обмеження надають можливість організувати систематичне виконання завдань здобувачами та не допустити значного накопичення незданих робіт на кінець семестру.

Модульна контрольна робота проводиться у вигляді тестових завдань, які дають можливість оцінити готовність здобувача до екзамену.

Заохоченням до своєчасного засвоєння теоретичного матеріалу є бали за участь у дискусійних обговореннях вивченого матеріалу на лекціях.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності здобувачі можуть бачити у відомості поточного контролю системи підтримки навчального процесу esatrus.kpi.ua. Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

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

Вказуються всі види контролю та бали за кожен елемент контролю, наприклад:

Поточний контроль: експрес-опитування, завдання комп'ютерного практикуму, модульна контрольна робота.

Семестровий контроль: екзамен

Умови допуску до семестрового контролю: семестровий рейтинг більше 50 балів.

Рейтинг студента з кредитного модуля складається з балів, які він отримує за результатами:

- 1) виконання модульних контрольних робіт;
- 2) виконання лабораторних комп'ютерних практикумів;
- 3) відповідей на екзаменаційні питання;
- 4) виконання практичного завдання екзаменаційного білету.

Система рейтингових балів

Усі завдання студента оцінюються за 100-бальною шкалою. Оцінка за завдання враховуються в семестрову оцінку студента з ваговим коефіцієнтом. Теоретичні знання студента оцінюють за результатами опитування на лекційних заняттях. Сумарна оцінка визначається за формулою:

$$T=0,2\cdot A+0,8\cdot (M_1+ M_2),$$

де A – додаткові бали за результатами дискусійного обговорення лекційного матеріалу, M_j – оцінка за j -ту модульну контрольну.

Практичні навички студента оцінюються за результатами захисту виконаних завдань комп'ютерного практикуму. Максимальна оцінка за виконання комп'ютерного практикуму, якщо він виконаний невчасно (тобто з більше, ніж тижневою затримкою) знижується на 10%. Сумарна оцінка визначається за формулою:

$$P = 1/13 \sum D_i,$$

де D_i – оцінка за i -тий комп'ютерний практикум.

Сума балів, набраних студентом протягом семестру, складається з сумарної оцінки за комп'ютерний практикум за формулою:

$$Z = 0,5 \cdot P + 0,5 \cdot T$$

де P – оцінка практичних навичок студента, T – оцінка його теоретичних знань.

Якщо набрана сума балів студента не влаштовує, йому пропонується виконання семестрової контрольної роботи або семестрового контрольного завдання в присутності викладача.

Екзаменаційна оцінка складається з оцінки за відповіді на теоретичні питання та виконане практичне завдання за формулою:

$$E = 0,5 \cdot P + 0,25 \cdot T_1 + 0,25 \cdot T_2$$

де P – оцінка виконання практичного завдання, T_1 , T_2 – оцінки відповідей на перше та друге теоретичні питання відповідно.

Підсумкова оцінка формується за результатами оцінювання знань та навичок студента в семестрі та на екзамені за формулою:

$$S = 0,5 \cdot Z + 0,5 \cdot E$$

Підсумкова оцінка переводиться до залікової оцінки згідно з таблицею:

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

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у Додатку А.

Робочу програму навчальної дисципліни (силабус):

Складено професор кафедри ІПІ, д.т.н., професор Стеценко Інна Вячеславівна

Ухвалено кафедрою ІПІ (протокол № 16 від 23.06.2025р.)

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