



ФОРМАЛЬНІ МЕТОДИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Робоча програма навчальної дисципліни (Силабус)

1. 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>4 кредити (120 год., з них 13 год. лекції, 26 год. комп'ютерний практикум, 81 год. самостійна робота)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен / -</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@ill.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/1lclgC5cab2Wil9SPbg9eOhgV0dFoXjt

2. 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 алгебри процесів (process algebras) та їх використання для дослідження взаємодії процесів,
- методи дослідження процесів (process mining),
- 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,
- сучасні тенденції розвитку формальних методів програмної інженерії

Студент повинен **вміти**:

- складати формалізовані моделі програмних систем з використанням алгебри процесів,
- застосовувати методи алгебри процесів для дослідження взаємодії обчислювальних процесів,
- 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,
- 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**:

ФК2 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.

ФК4 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.

ФК 6 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:

ПРН1 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.

ПРН3 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.

ПРН6 Розуміти теоретичні засади, що лежать в основі методів досліджень інформаційних систем та програмного забезпечення, методології проведення досліджень та обчислювальних експериментів.

ПРН11 Уміти створювати технологічне середовище комп'ютерної моделі інформаційної системи.

ПРН13 Знати методології комп'ютерного моделювання складних систем.

ПРН14 Знати методології візуального моделювання складних систем.

ПРН16 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», а також використовуються в розробці наукової дисертації доктора філософії.

3. Зміст навчальної дисципліни

Розділ 1 Формалізм мереж Петрі.

Тема 1.1. Огляд формальних методів інженерії програмного забезпечення.

Тема 1.2. Класичні мережі Петрі.

Тема 1.3. Стохастичні мережа Петрі.

Тема 1.4. Моделювання паралельних обчислень з використанням базових та стохастичних мереж Петрі, Петрі-об'єктних моделей.

Розділ 2 Алгебри процесів.

Тема 2.1. Формальні методи алгебри процесів.

Тема 2.2. Формалізм Мілнера для опису взаємодії систем. π -числення: основні поняття та правила виведення.

4. Навчальні матеріали та ресурси

Базова література

3. Hoare C.A.R., Jifeng H. *Unifying Theories of Programming*. New York: Prentice-Hall, 1998. — 311 p.

4. Jensen K. *Coloured Petri Nets: Modeling and Validation of Concurrent Systems* / K.Jensen, L.Kristensen - Springer-Verlag Berlin Heidelberg, 2000 – 383p.

5. Стеценко И.В. Теоретические основы Петри-объектного моделирования систем / И.В. Стеценко // Математичні машини і системи. – Київ, 2011. - №4. – С.136-148.

6. 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. ICCSEEA 2019. Advances in Intelligent Systems and Computing*, vol 938, 78-89. Springer, Cham. 2020.

7. 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. ICCSEEA 2018. Advances in Intelligent Systems and Computing*, vol 754, 391-401. Springer, Cham. 2020.

8. 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.

9. Nicola R. A gentle introduction to Process Algebras <https://pdfs.semanticscholar.org/12d9/eae1638729aeb237b5be445ee91ecdd3c5d7.pdf>

10. Baeten J.C.M., Beek D.A., Rooda J.E. Process algebra <http://mate.tue.nl/mate/pdfs/8509.pdf>

Юрчишин В.М., Піх В.Я. Формальні методи програмної інженерії: конспект лекцій. – Івано-Франківськ: ІФТУНГ, 2016. – 154 с.

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.*

11. Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
Розділ 1 Формалізм мереж Петрі.		
1.	Лекція 1.1 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 1.1 UML-опис програмного забезпечення	Завдання: – Визначити переваги та недоліки опису програмного забезпечення засобами UML. – Виконати аналіз властивостей програмного забезпечення на основі його UML-опису.
3.	Лекція 1.2 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 1.2 Математичне моделювання процесів передачі даних класичною мережею Петрі	Завдання: – Розробити модель системи передачі даних класичною мережею Петрі. – Скласти математичний опис моделі. – Дослідити властивості мережі Петрі: обмеженість, досяжність, консервативність, циклічність.
5.	Комп'ютерний практикум 1.3 Імітаційне моделювання процесів передачі даних класичною мережею Петрі	Завдання: – Розробити імітаційну модель передачі даних телекомунікаційної системи – Використовуючи імітаційну модель дослідити можливість виникнення тупикової ситуації в системі передачі даних
6.	Лекція 1.3 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 CPNTools. – Петрі-об'єктна модель: визначення, математичний опис, фундаментальні твердження, математичні рівняння станів. Програмне забезпечення PetriObjModelPaint.
7.	Комп'ютерний практикум 1.4 Математичне моделювання обчислювальних процесів стохастичною мережею Петрі	Завдання: – Розробити модель веб застосування стохастичною мережею Петрі. – Скласти математичні рівняння станів стохастичної мережі Петрі.
8.	Комп'ютерний практикум 1.5 Імітаційне моделювання обчислювальних процесів стохастичною мережею Петрі	Завдання: – Розробити імітаційну модель веб застосування. – Дослідити ефективність обчислювальних процесів веб застосування використовуючи імітаційну модель. – Порівняти результати, отримані на моделі, з результатами експериментального дослідження веб застосування
9.	Лекція 1.4 Моделювання паралельних обчислень з використанням базових 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.
10.	Комп'ютерний практикум 1.6. 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.
11.	Комп'ютерний практикум 1.7. Імітаційне моделювання паралельного алгоритму 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
12.	Комп'ютерний практикум 1.8. 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.

13.	Комп'ютерний практикум 1.9. 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.
14.	Модульна контрольна робота	
Розділ 2 Алгебри процесів.		
15.	Лекція 2.1. Формальні методи алгебри процесів.	– Формальний опис обчислювальних процесів. – Поняття про еквівалентність процесів. – Сучасні тенденції розвитку формальних методів алгебри процесів. – Теорія взаємодіючих послідовних процесів Хоара. Логіка Хоара.
16.	Комп'ютерний практикум 2.1.	Завдання: – Розробити модель взаємодії процесів, що розглядались у попередніх практикумах, використовуючи логіку Хоара. – Виконати аналіз взаємодії процесів, використовуючи логіку Хоара.
17.	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: Хоара, Мілнера, Бергстра і Клопа. – Формальний опис псевдопаралельних процесів з використанням алгебри процесів. – Парадигми програмування та формалізми опису обчислювальних процесів.
18.	Комп'ютерний практикум 2.2	Завдання: – Розробити модель взаємодії процесів, що розглядались у попередніх практикумах, використовуючи алгебру процесів Мілнера. – Виконати аналіз взаємодії процесів, використовуючи алгебру процесів Мілнера
19.	Модульна контрольна робота	

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді на гугл-диску викладача, до якого надано доступ групі здобувачів, а також, в середовищі Moodle платформи дистанційного навчання «Сікорський». Контент платформи доступний із будь-якого місця в мережі Інтернет зареєстрованому на курс студенту. Навчальний курс в середовищі Moodle 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 із використанням сучасних мультимедійних презентаційних технологій. На лекціях проводяться експрес-опитування, які дають можливість лектору отримати інформацію про якість засвоєння матеріалу та, за необхідності, розглянути

більш детально складний матеріал. Обговорення лекційного матеріалу відбувається також в чаті дисципліни в середовищі Moodle.

Лабораторні заняття (Комп'ютерні практикуми) виконуються з використанням мови програмування Java, програмного забезпечення CPNTools, що надається безкоштовно в навчальних цілях а також програмного забезпечення PetriObjPaint, розміщеного в github.com/StetsenkoInna. Під час їх проведення використовуються методичні вказівки до комп'ютерних практикумів по дисципліні. 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.

Модульні контролі проводяться за результатами виконання комп'ютерних практикумів та модульних контрольних робіт. Модульні контрольні роботи містять завдання з перевірки як теоретичних знань, так і практичних навичок.

В умовах дистанційного навчання 2021-2022 н.р. усі види занять, у тому числі контрольні заходи, проводяться з використанням Google Meet.

6. Самостійна робота студента/аспіранта

Матеріали для самостійного вивчення дисципліни розміщені викладачем в електронному вигляді на гугл-диску викладача, до якого надано доступ групі студентів та асистентам, які ведуть заняття комп'ютерного практикуму, а також, в середовищі Moodle платформи дистанційного навчання «Сікорський». 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.

12. 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 відповідному розділі дисципліни у мудл платформи «Сікорський». Несвоєчасним вважається виконання завдання з затримкою більше 1 тижня і штрафується втратою 10% оцінки. 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.

Модульні контролі проводяться за результатами виконання комп'ютерних практикумів та модульних контрольних робіт. Файли з виконаними завданнями модульних контрольних робіт розміщуються студентами у відповідному розділі дисципліни у мудл платформи «Сікорський».

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, МКР, завдання комп'ютерного практикуму

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

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.

Враховуються також додаткові бали за правильні відповіді на запитання під час лекційних занять.

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 письмових модульних контрольних робіт та, частково, за результатами опитування на лекційних заняттях. При невчасному виконанні модульної контрольної роботи її оцінка знижується на 10%. Сумарна оцінка визначається за формулою:

$$T=0,5 \cdot K1+0,5 \cdot K2+A,$$

де $K1$ – оцінка за першу модульну контрольну роботу, $K2$ – оцінка за другу модульну контрольну роботу, A – додаткові бали за результатами обговорення на лекційних заняттях.

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:

$$P=1/18 \sum D_i,$$

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:

$$E = 0,5 \cdot P+0,25 \cdot T1+0,25 \cdot T2$$

де P – оцінка виконання практичного завдання, $T1$, $T2$ – 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 АСОІУ, д.т.н., професор Стеценко Інна Вячеславівна

Ухвалено кафедрою АСОІУ (протокол № від . 06.2023р.)

Погоджено Методичною комісією факультету (протокол № від . 06.2023р.)