

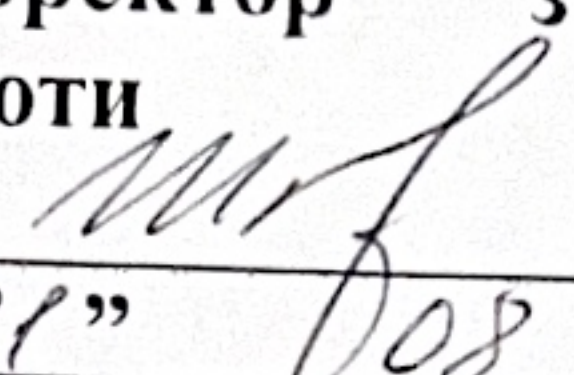
ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з навчально-методичної
роботи


“31” 008 2026 р. Ярослав ШТАНЬКО

Паралельні та розподілені обчислення

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	перший (бакалаврський)
Статус дисципліни:	обов'язкова
Мова викладання, навчання та оцінювання:	українська
Семестр викладання	V

Івано-Франківськ
2026

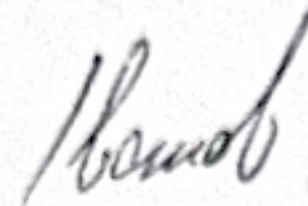
РОЗРОБНИК:
викладачка кафедри ІТ



Микола ДЕМЧИНА

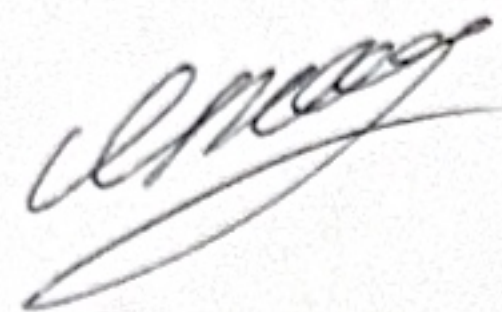
ЗАТВЕРДЖЕНО:
на засіданні кафедри інформаційних технологій, протокол № 1 від 27.08.2026 р.

Завідувач кафедри
УЗГОДЖЕНО:



Олександр ІВАНОВ

Гарант ОПП/ОНП



Володимир МАКОВИШИН

СХВАЛЕНО:
на засіданні Науково-методичної ради, протокол № 1 від 31.08.2026 р.

е-mail	mykola.demchyna@ukd.edu.ua
Номер аудиторії чи кафедри	206
Посилання на сайт	https://ukd.edu.ua
Сторінка курсу в СДО	https://online.ukd.edu.ua/course/view.php?id=3788

ВСТУП

Анотація навчальної дисципліни

Навчальна дисципліна «Паралельні та розподілені обчислення» охоплює архітектури паралельних систем, організацію обчислень у середовищах зі спільною та розподіленою пам'яттю, а також кластерні й GRID-системи. Розглядаються паралелізм за задачами та даними, закон Амдала, технології OpenMP, MPI, CUDA, а також механізми синхронізації процесів.

Навчання проводиться у формі лекцій і практичних занять. Практична підготовка зосереджена на багатопотоковому програмуванні мовою Java: створенні потоків, керуванні їхнім життєвим циклом, захисті спільних даних, міжпотоківій взаємодії та використанні пулів потоків. Здобувачі перевіряють коректність програм і вимірюють ефективність паралельного виконання.

Мета навчальної дисципліни — сформувати знання про архітектури паралельних і розподілених систем та засоби синхронізації процесів, а також практичні навички розроблення, налагодження й оцінювання багатопотокових застосунків.

Основні завдання навчальної дисципліни

- опанувати принципи організації паралельних обчислень і вибору моделі розпаралелювання;
- навчитися добирати механізми синхронізації та запобігати помилкам спільного доступу до даних;
- розробляти й перевіряти багатопотокові програми та обґрунтовувати їх ефективність.

У результаті вивчення дисципліни здобувач повинен знати

- архітектурні класи паралельних систем, принципи роботи кластерів і GRID-середовищ;
- моделі паралельного виконання програм, показники прискорення й ефективності;
- критичні секції, взаємне виключення, семафори, монітори та бар'єри;
- основи OpenMP, MPI, CUDA, POSIX Threads і засоби багатопотоковості Java.

У результаті вивчення дисципліни здобувач повинен уміти

- аналізувати архітектурні особливості обчислювальних систем і розпаралелювати задачі;
- реалізовувати синхронізацію процесів і потоків, зокрема за допомогою семафорів та бар'єрів;
- розробляти багатопотокові програми мовою Java та використовувати потокобезпечні структури даних;
- налагоджувати й тестувати паралельні програми, виявляти гонки даних і взаємні блокування;
- вимірювати прискорення, оцінювати масштабованість та обґрунтовувати вибір програмних засобів.

Компетентності та результати навчання, яких набувають здобувачі освіти внаслідок вивчення навчальної дисципліни (шифри та зміст компетентностей та програмних результатів навчання вказано відповідно до ОПП “Розробка та тестування програмного забезпечення”.

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
K14. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування K15. Здатність розробляти архітектури, модулі та компоненти програмних систем K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами K19. Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя K26. Здатність до алгоритмічного та логічного мислення	ПР05. Знати і застосовувати відповідні методи доменного, системного і об’єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення ПР12. Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	3		
Семестр	5		
Кількість кредитів ECTS	6		
Аудиторні навчальні заняття		денна форма	заочна форма
	лекції	32	8
	практичні	28	8
Самостійна робота		120	164
Форма підсумкового контролю	Екзамен		

Структурно-логічна схема вивчення навчальної дисципліни

Пререквізити	Постреквізити
Основи програмування; Алгоритми та структури даних; Об'єктно-орієнтоване програмування.	Системне програмування; Якість програмного забезпечення і тестування.

Для практичних занять потрібні навички програмування мовою Java, роботи з класами, колекціями та оброблення винятків. Використовуються комп'ютер із JDK, середовище розроблення, засоби налагодження та доступ до курсу в СДО.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Перелік тем лекційного матеріалу

Години наведено для денної та заочної форм навчання. У записі годин самостійного вивчення перше число стосується денної, друге — заочної форми. Ці години входять до загального обсягу самостійної роботи.

Тема 1. Основи паралельних обчислень та архітектура систем

Лекції: денна форма — 4 год.; заочна форма — 1 год.

Визначення і переваги паралелізму. Класифікація комп'ютерних систем. Векторно-конвеєрні архітектури, мультипроцесори зі спільною пам'яттю та мультикомп'ютери з розподіленою пам'яттю.

Питання для самостійного вивчення (4 год. / 6 год.): Основні концепції паралелізму; порівняння архітектур зі спільною та розподіленою пам'яттю. Література: [1, 3, 4].

Тема 2. Кластерні та GRID-системи

Лекції: денна форма — 4 год.; заочна форма — 1 год.

Архітектури кластерів. Концепція GRID і метакомп'ютинг. Взаємодія вузлів через обмін повідомленнями. Масштабування, затримки в мережі та балансування навантаження в розподілених системах.

Питання для самостійного вивчення (4 год. / 6 год.): Класифікація мультипроцесорних систем; принципи організації кластерів; відмінності між кластерними та GRID-обчисленнями. Література: [1, 3, 4].

Тема 3. Основні концепції паралелізму та програмні підходи

Лекції: денна форма — 6 год.; заочна форма — 2 год.

Паралелізм за задачами і даними. Декомпозиція задач, залежності між підзадачами та розподіл робіт. Закон Амдала. Прискорення, ефективність і накладні витрати паралельного виконання.

Питання для самостійного вивчення (4 год. / 6 год.): Типи паралелізму; обмеження прискорення за законом Амдала; стратегії розпаралелювання та способи порівняння їхньої ефективності. Література: [1, 4].

Тема 4. Технології та інструменти паралельного програмування

Лекції: денна форма — 6 год.; заочна форма — 2 год.

Апаратний і програмний паралелізм. GPGPU. Призначення, моделі пам'яті та базові засоби OpenMP, MPI і CUDA. Вибір технології відповідно до архітектури системи та характеру задачі.

Питання для самостійного вивчення (6 год. / 8 год.): Можливості GPGPU; приклади застосування OpenMP, MPI і CUDA; порівняння моделей спільної пам'яті, обміну повідомленнями та виконання на GPU. Література: [1, 6, 7, 8].

Тема 5. Синхронізація процесів і критичні секції

Лекції: денна форма — 6 год.; заочна форма — 1 год.

Гонки даних і постановка задачі критичної секції. М'ютекси, семафори, монітори та бар'єри. Алгоритми взаємного виключення, зокрема алгоритм квитка та алгоритм поліклініки. Взаємне блокування та голодування.

Питання для самостійного вивчення (4 год. / 6 год.): Умови коректного взаємного виключення; застосування семафорів і бар'єрів; аналіз алгоритмів і запобігання взаємним блокуванням. Література: [1, 2, 5, 9].

Тема 6. Методи багатопотокового програмування

Лекції: денна форма — 6 год.; заочна форма — 1 год.

Основи POSIX Threads. Створення та життєвий цикл потоків Java. Блокування й потокобезпечні колекції. Видимість змін у пам'яті та атомарність операцій. Пули потоків і засоби налагодження багатопотокових програм.

Питання для самостійного вивчення (4 год. / 6 год.): Thread і Runnable; синхронізація в Java; атомарні операції; потокобезпечні структури даних; порівняння Java та POSIX Threads. Література: [1, 2, 5, 9].

Разом лекцій: денна форма — 32 год.; заочна форма — 8 год.

Зміст практичних занять

Тема 1. Створення та запуск потоків у мові Java

Практичні заняття: денна форма — 4 год.; заочна форма — 1 год.

Реалізація потоків через Thread і Runnable, анонімні класи та лямбда-вирази. Запуск кількох задач, керування головним потоком і порівняння послідовного та паралельного виконання.

Питання для самостійного вивчення (2 год. / 2 год.): Способи створення потоків; відмінності між наслідуванням Thread і реалізацією Runnable. Література: [2, 5, 9].

Тема 2. Керування життєвим циклом потоків

Практичні заняття: денна форма — 4 год.; заочна форма — 1 год.

Дослідження станів NEW, RUNNABLE, WAITING, TIMED_WAITING, BLOCKED і TERMINATED. Використання sleep(), join(), yield(), interrupt(). Пріоритети та потоки-демони. Коректне завершення роботи.

Питання для самостійного вивчення (2 год. / 2 год.): Станові діаграми потоків; призначення sleep(), join(), yield(); потоки-демони та оброблення переривань. Література: [2, 5, 9].

Тема 3. Синхронізація потоків у Java

Практичні заняття: денна форма — 6 год.; заочна форма — 2 год.

Відтворення гонки даних. Захист критичних секцій за допомогою synchronized і ReentrantLock. Порівняння моніторів та явних блокувань. Перевірка коректності спільного лічильника та аналіз взаємного блокування.

Питання для самостійного вивчення (4 год. / 4 год.): Монітори та м'ютекси; synchronized для методів і блоків; можливості ReentrantLock і правила звільнення блокувань. Література: [2, 5, 9].

Тема 4. Міжпотоківі комунікації

Практичні заняття: денна форма — 6 год.; заочна форма — 2 год.

Робота з wait(), notify(), notifyAll() та Condition. Реалізація задачі «виробник–споживач» через монітор і BlockingQueue. Застосування семафорів і бар'єрів для координації задач.

Питання для самостійного вивчення (4 год. / 6 год.): Очікування умов у циклі; відмінності низькорівневих засобів взаємодії та BlockingQueue; сценарії застосування семафорів і бар'єрів. Література: [2, 5, 9].

Тема 5. Додаткові засоби синхронізації та пули потоків

Практичні заняття: денна форма — 8 год.; заочна форма — 2 год.

Видимість даних у volatile та обмеження його застосування; final і незмінні об'єкти; атомарні класи. ExecutorService, Callable, Future. Створення пулу, отримання результатів, оброблення помилок і завершення роботи. Порівняння часу виконання для різних кількостей потоків.

Питання для самостійного вивчення (4 год. / 6 год.): Відмінність видимості від атомарності; гарантії volatile і final; добір пулу потоків; асинхронне виконання та методика вимірювання ефективності. Література: [2, 5, 9].

Разом практичних занять: денна форма — 28 год.; заочна форма — 8 год.

За кожною практичною темою здобувач подає програмний код, короткий звіт із результатами перевірки та висновками і захищає виконану роботу. Для заочної форми частина реалізації та підготовка звіту виконуються самостійно; на заняттях проводяться консультація, перевірка і захист.

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни:

Найменування видів робіт	Розподіл годин	
	денна форма	заочна форма
Самостійна робота, год, у т.ч.:	120	164
Опрацювання матеріалу, викладеного на лекціях	19	26
Підготовка до практичних занять та підсумкового контролю	30	41
Підготовка звітів з практичних робіт	19	26
Підготовка до поточного контролю	10	13
Опрацювання матеріалу, винесеного на самостійне вивчення	42	58

Питання для самостійного вивчення після лекційних тем становлять 26 годин для денної та 38 годин для заочної форми; після практичних тем — відповідно 16 і 20 годин. Разом — 42 і 58 годин у межах останнього рядка таблиці. Інші години охоплюють опрацювання лекцій, підготовку програм, звітів і контрольних заходів.

Результати самостійної роботи перевіряються під час обговорення тем, захисту практичних робіт, поточного тестування та екзамену. Здобувач має пояснювати прийняті рішення й демонструвати відтворювані результати виконання програм.

ПОЛІТИКА КУРСУ

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу здобувачів освіти». Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).

Здобувачі мають право оскаржувати результати поточного оцінювання, а також інформувати про факти конфліктних ситуацій в академічній групі чи з викладачем у відповідності до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку), якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за [покликанням](#).



Вимоги щодо дотримання академічної доброчесності, види її порушень і порядок притягнення до академічної відповідальності визначено Положенням про систему забезпечення академічної доброчесності в Університеті Короля Данила. Ознайомитись з документом можна за [покликанням](#).

Використання систем штучного інтелекту під час виконання навчальних завдань допускається відповідно до “Політики використання штучного інтелекту в Університеті Короля Данила”. Здобувач зобов’язаний критично перевіряти створений ШІ зміст, нести відповідальність за подану роботу та зазначати факт і спосіб використання таких систем. Ознайомитись з документом можна за [покликанням](#).



Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання. Ознайомитись з відповідним документом можна за [покликанням](#).



Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [покликанням](#).



МЕТОДИ НАВЧАННЯ

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця в галузі інформаційних технологій.

Результат навчання	Метод навчання	Метод оцінювання
ПР05. Знати і застосовувати відповідні методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення	Лекція, моделювання структури задачі, аналіз залежностей і розрахунок прискорення.	Розрахункові завдання, пояснення моделі розпаралелювання, екзамен.
ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення	Порівняння моделей паралелізму, проблемні завдання, вибір технології для заданої архітектури.	Обґрунтування вибору технології, поточне опитування, екзамен.
ПР12. Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення	Практичне програмування, аналіз і вдосконалення багатопотокових рішень.	Виконання та захист практичних робіт, аналіз коду, екзамен.
ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних	Демонстрування обміну даними, вправи зі спільними колекціями та міжпоточною взаємодією.	Перевірка програм, захист задачі «виробник–споживач», екзамен.
ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення	Тестування, налагодження, відтворення гонок даних і взаємних блокувань.	Аналіз тестів та їх результатів, захист звітів, екзамен.

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання здійснюється відповідно до «Положення про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти» у редакції від 01.07.2026. [Положення УКД](#)

Денна форма навчання

Вид	Зміст	% від загальної оцінки	Бал	
			min	max
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	екзамен	40	25	40
Всього:	-	100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація поточного контролю здійснюється в «Електронному журналі обліку успішності академічної групи» на підставі чотирибальної шкали - “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення дисципліни впродовж наступного навчального семестру.

За результатами підсумкового контролю (диференційований залік/екзамен) студент може отримати 40 балів.

Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен/диференційований залік і повинні йти на перездачу.

Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно Шкали оцінювання знань за ЄКТС) і є сумою балів отриманих під час поточного та підсумкового контролю.

Підсумковий контроль з дисципліни Паралельні та розподілені обчислення проводиться у вигляді Екзамен.

Шкала оцінювання знань за ЄКТС

Оцінка за національною шкалою	Рівень досягнень, бали	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна література

1. Демчина М. М. Паралельне програмування : конспект лекцій. Івано-Франківськ : ІФНТУНГ, 2015. 176 с.
2. Демчина М. М. Паралельне програмування : лабораторний практикум. Івано-Франківськ : ІФНТУНГ, 2016. 44 с.
3. Pacheco R., Malensek M. An Introduction to Parallel Programming. 2nd ed. Morgan Kaufmann, 2022. URL: <https://www.cs.usfca.edu/~peter/ipp2/index.html> (дата звернення: 13.09.2026).
4. Goetz B., Peierls T., Bloch J., Bowbeer J., Holmes D., Lea D. Java Concurrency in Practice. Addison-Wesley, 2006. URL: <https://jcip.net/> (дата звернення: 13.09.2026).
5. Van Steen M., Tanenbaum A. S. Distributed Systems. 4th ed. distributed-systems.net, 2023. URL: <https://www.distributed-systems.net/index.php/books/ds4/> (дата звернення: 13.09.2026).
6. Foster I. Designing and Building Parallel Programs. Addison-Wesley, 1995. URL: <https://www.mcs.anl.gov/~itf/dbpp/> (дата звернення: 13.09.2026).

Електронні інформаційні ресурси

7. Oracle. Java Platform, Standard Edition 21 API Specification. Package java.util.concurrent. URL: <https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/concurrent/package-summary.html> (дата звернення: 13.09.2026).
8. OpenMP Architecture Review Board. OpenMP API Specifications. URL: <https://www.openmp.org/specifications/> (дата звернення: 13.09.2026).
9. MPI Forum. MPI Documents. MPI 5.0 Standard, 2025. URL: <https://www.mpi-forum.org/docs/> (дата звернення: 13.09.2026).
10. NVIDIA. CUDA Programming Guide. URL: <https://docs.nvidia.com/cuda/cuda-programming-guide/> (дата звернення: 13.09.2026).
11. Oracle. The Java Tutorials. Lesson Concurrency. Навчальні приклади базових механізмів Java. URL: <https://docs.oracle.com/javase/tutorial/essential/concurrency/> (дата звернення: 13.09.2026).