

Чернівецький національний університет імені Юрія Федьковича

Факультет математики та інформатики

Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”

Декан факультету математики та інформатики

_____ Ольга МАРТИНЮК

30 червня 2026 року

РОБОЧА ПРОГРАМА

навчальної дисципліни

Паралельне програмування

Обов’язкова навчальна дисципліна

Освітньо-професійна програма	<u>Технології програмування та комп’ютерне моделювання</u>
Спеціальність	<u>F1 Прикладна математика</u>
Галузь знань	<u>F Інформаційні технології</u>
Рівень вищої освіти	<u>другий (магістерський)</u>
Факультет	<u>математики та інформатики</u>
Мова навчання	<u>українська</u>

Чернівці, 2026 рік

Робоча програма навчальної дисципліни "Паралельне програмування" **складена відповідно до** освітньо-професійної програми "Технології програмування та комп'ютерне моделювання" другого (магістерського) рівня вищої освіти за спеціальністю "F1 Прикладна математика" галузі знань "F Інформаційні технології", затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича 28 квітня 2025р., протокол №5.

Розробник:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Викладач, що забезпечує читання даної навчальної дисципліни:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Погоджено з гарантом ОП _____ Микола ФІЛПЧУК

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій
Протокол № 13 від 23 червня 2026 року
Завідувач кафедри _____ **Ярослав БІГУН**

Схвалено методичною радою факультету математики та інформатики
Протокол № 12 від 30 червня 2026 року
Голова методичної ради _____ **Тоня ФРАТАВЧАН**

Затверджено Вченою радою факультету математики та інформатики
Протокол № 15 від 30 червня 2026 року
Голова Вченої ради _____ **Ольга МАРТИНЮК**

МЕТА ОСВІТНЬОГО КОМПОНЕНТА

Мета дисципліни:

- формування у студентів необхідного рівня теоретичної і практичної підготовки для організації ними ефективного розв'язку великих задач на обчислювальних системах з паралельною архітектурою з використанням сучасних пакетів та стандартів паралельного програмування;
- оволодіння методами та технологіями побудови паралельних алгоритмів;
- набуття практичних навичок гетерогенного програмування;
- вивчення методів оптимізації програм для паралельних обчислювальних систем та забезпечення вміннями узгодження потенціального паралелізму алгоритму з наявним апаратним забезпеченням.

Завдання дисципліни:

- вивчення стандартів паралельного програмування (багатопотоковість, обмін повідомленнями між процесами, гетерогенні обчислення) та їх реалізацій;
- вивчення технологій паралельного програмування (CUDA, OpenCL), паралельних числових методів рішення задач з метою використання набутих знань для розробки паралельних алгоритмів та створення паралельних програм;
- отримання знань і умінь для розробки паралельних алгоритмів з метою виконання на GPU;
- засвоєння знань, умінь та навичок гетерогенного програмування.

Знання, які студент отримає в результаті вивчення даної дисципліни, відіграватимуть важливу роль у процесі його професійного формування та зростання.

ПРЕРЕКВІЗИТИ

Для успішного засвоєння дисципліни необхідні добрі знання класичного програмування на CPU мовами C, C++, Java, Python, JavaScript, C#, що закладаються під час навчання в бакалавраті в навчальних курсах, пов'язаних з вивченням мов програмування.

РЕЗУЛЬТАТИ НАВЧАННЯ

В результаті вивчення дисципліни студент має набути таких **компетентностей**:

знати:

- мови та спеціальні бібліотеки паралельного програмування (CUDA, OpenCL та ін.);
- принципи організації неграфічних обчислень на графічних процесорах;
- специфіку розробки багатопотокових додатків;
- проблеми синхронізації та засоби їх вирішення при паралельному програмуванні;
- критерії вибору програмно-апаратної платформи для вирішення обчислювально-складних задач заданого класу;

вміти:

- проводити декомпозицію поставленого завдання на менші частини, які можуть бути виконані одночасно і незалежно одна від одної, з метою побудови паралельного алгоритму заданої задачі;
- проводити оптимізацію паралельних алгоритмів;
- розробляти ефективні паралельні програми;
- створювати багатопотокові додатки та задіювати систему планування виконання потоків;
- самостійно розробляти схему взаємодії процесів, обирати оптимальні засоби реалізації взаємодії процесів;
- працювати на паралельних комп'ютерах та кластерах;
- виконувати гетерогенні обчислення (CPU та GPU);
- використовувати принципи роботи з пам'яттю та потоками в програмах на CUDA;
- задіювати засоби синхронізації паралельних обчислень без втрати продуктивності.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності;

ЗК6. Здатність вчитися і оволодівати сучасними знаннями, інформаційними технологіями та комп'ютерною технікою;

фахові компетентності:

ФК9. Здатність проектувати та розробляти програмне забезпечення для розв'язання формалізованих задач із застосуванням різних парадигм програмування, зокрема об'єктно-орієнтованого, методів паралельного програмування та захисту даних

та отримуються наступні **програмні результати навчання:**

ПРН9. Створювати та програмно реалізовувати алгоритми розв'язання задач, розробляти системне та прикладне забезпечення інформаційних систем і технологій з урахуванням вимог до його якості, надійності, виробничих характеристик;

ПРН10. Аргументувати вибір програмних та технічних засобів для розробки програмних засобів на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог замовника; мати навички налагодження та тестування програмних і технічних засобів;

ПРН11. Вміти проектувати архітектуру системи з великими обсягами даних.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	5	10	4	120	15	-	-	30	75	-	іспит
Заочна	-	-	-	-	-	-	-	-	-	-	-

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						Заочна форма						
	усього	у тому числі											
		л	п	лаб	інд	с.р.							
1	2	3	4	5	6	7							
Змістовий модуль 1. <i>Архітектура GPU та основи гетерогенного програмування GPGPU</i>													
Тема 1. Архітектура та структура пам'яті GPU	10	1		3		6							
Тема 2. Реалізація масивного паралелізму за допомогою графічних процесорів	10	1		3		6							
Тема 3. Принципи організації паралельних обчислень	10	1		3		6							
Тема 4. Гетерогенне програмування GPGPU	10	1		3		6							

Разом за змістовим модулем 1	40	4		12		24						
Змістовий модуль 2. Технологія CUDA від компанії NVIDIA												
Тема 1. Вступ до CUDA: архітектура та модель програмування	10	1		2		7						
Тема 2. Основи написання CUDA-ядер	10	2		2		6						
Тема 3. Управління пам'яттю в CUDA	10	2		2		6						
Тема 4. Оптимізація продуктивності CUDA-програм	10	1		2		7						
Тема 5. Розробка паралельних алгоритмів з використанням CUDA та додаткових бібліотек	10	1		2		7						
Разом за змістовим модулем 2	50	5		10		33						
Змістовий модуль 3. Гетерогенне програмування за допомогою OpenCL												
Тема 1. Реалізація масивного паралелізму за допомогою графічних процесорів	9	1		2		6						
Тема 2. Написання та виконання OpenCL-ядер	7	1		2		4						
Тема 3. Оптимізація OpenCL-програм: продуктивність та налагодження	7	1		2		4						
Тема 4. Гетерогенні обчислення з	7	1		2		4						

OpenCL: використання CPU, GPU та їх комбінація												
Разом за змістовим модулем 3	30	4		8		16						
Усього годин	120	15		30		75						

Тематика лекційних занять

МОДУЛЬ 1. АРХІТЕКТУРА GPU ТА ОСНОВИ ГЕТЕРОГЕННОГО ПРОГРАМУВАННЯ GPGPU	
Тема 1	Архітектура та структура пам'яті GPU
Тема 2	Реалізація масивного паралелізму за допомогою графічних процесорів
Тема 3	Принципи організації паралельних обчислень
Тема 4	Гетерогенне програмування GPGPU
МОДУЛЬ 2. ТЕХНОЛОГІЯ CUDA ВІД КОМПАНІЇ NVIDIA	
Тема 1	Вступ до CUDA: архітектура та модель програмування
Тема 2	Основи написання CUDA-ядер
Тема 3	Управління пам'яттю в CUDA
Тема 4	Оптимізація продуктивності CUDA-програм
Тема 5	Розробка паралельних алгоритмів з використанням CUDA та додаткових бібліотек
МОДУЛЬ 3. ГЕТЕРОГЕННЕ ПРОГРАМУВАННЯ ЗА ДОПОМОГОЮ OPEN CL	
Тема 1	Реалізація масивного паралелізму за допомогою графічних процесорів
Тема 2	Написання та виконання OpenCL-ядер
Тема 3	Оптимізація OpenCL-програм: продуктивність та налагодження
Тема 4	Гетерогенні обчислення з OpenCL: використання CPU, GPU та їх комбінація

Тематика семінарських занять

Семінарські заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика практичних занять

Практичні заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика лабораторних занять

МОДУЛЬ 1. АРХІТЕКТУРА GPU ТА ОСНОВИ ГЕТЕРОГЕННОГО ПРОГРАМУВАННЯ GPGPU	
ЛР 1.1	Реалізація масивного паралелізму за допомогою графічних процесорів
ЛР 1.2	Гетерогенне програмування GPGPU
МОДУЛЬ 2. ТЕХНОЛОГІЯ CUDA ВІД КОМПАНІЇ NVIDIA	
ЛР 2.1	Вступ до CUDA: архітектура та модель програмування
ЛР 2.2	Основи написання CUDA-ядер
ЛР 2.3	Управління пам'яттю в CUDA
МОДУЛЬ 3. ГЕТЕРОГЕННЕ ПРОГРАМУВАННЯ ЗА ДОПОМОГОЮ OPEN CL	
ЛР 3.1	Написання та виконання OpenCL-ядер
ЛР 3.2	Гетерогенні обчислення з OpenCL: використання CPU, GPU та їх комбінація

Індивідуальні науково-дослідні завдання (ІНДЗ)

ІНДЗ з даної навчальної дисципліни не передбачені.

Завдання для самостійної роботи студентів

Назва теми	Кількість годин
Порівняльний аналіз CPU та GPU для масивно-паралельних обчислень (архітектурні відмінності, сфери застосування, обмеження)	8
Порівняльний аналіз CPU та GPU для масивно-паралельних обчислень (архітектурні відмінності, сфери застосування, обмеження)	8
Оптимізація продуктивності CUDA-програм: типові помилки та кращі практики (occupancy, coalesced access, latency hiding)	14
Застосування GPGPU у реальних задачах (обробка зображень, машинне навчання, фізичне моделювання — на вибір студента)	14
Порівняння технологій CUDA та OpenCL для гетерогенних обчислень (портативність, продуктивність, складність розробки)	8
Сучасні тенденції розвитку гетерогенного програмування (multi-GPU, unified memory, heterogeneous computing у HPC та AI)	8
Разом	60

Самостійна робота студента полягає в опрацюванні лекційного матеріалу, більш детальному розгляді окремих питань курсу, підготовці до виконання та захисту лабораторних робіт.

Самостійна робота студента загалом складає 60 годин. Розподіл цих годин за видами робіт: опрацювання лекційного матеріалу, більш детальний розгляд окремих питань курсу – 30 годин; підготовка до виконання та захисту лабораторних робіт – 30 годин.

Оцінювання самостійної роботи студента є складовою частиною оцінювання захисту лабораторних робіт (30% оцінки кожної лабораторної роботи складає оцінювання самостійної роботи).

МЕТОДИ НАВЧАННЯ

Методи, що використовуються у освітньому процесі:

- методи формування професійної компетентності – пояснення, демонстрація, візуалізація;
- методи формування практичних умінь та навичок – розв’язування задач, виконання практичних завдань.

СИСТЕМА КОНТРОЛЮ ТА ОЦІНЮВАННЯ

Методами та формами поточного контролю є:

- усні поточні опитування;
- захист виконаних лабораторних робіт.

Формою підсумкового контролю є усний залік.

ПОЛІТИКА ОСВІТНЬОГО ПРОЦЕСУ

Політика щодо відвідування. Відвідування занять є обов’язковим. За наявності поважних причин (хвороба, індивідуальний графік навчання) навчання, за погодженням із викладачем, може відбуватись у змішаній (очно-дистанційній) формі. За необхідності, з метою з’ясування всіх незрозумілих під час самостійної роботи питань, потрібно відвідувати консультації викладача.

Політика щодо академічної доброчесності. Студент зобов’язаний самостійно, своєчасно та добросовісно виконувати усі отримані завдання, дотримуючись принципів академічної доброчесності. Під час проведення контрольних заходів заборонені використання сторонньої допомоги, навчальної літератури, конспектів, мобільних пристроїв, інших джерел інформації.

Політика щодо дедайнів та перескладання. Якщо студент був відсутній на заняттях (з будь-якої причини), він повинен самостійно опрацювати пропущений матеріал та прозвітувати викладачу про виконання відповідних завдань у встановлені ним терміни під час консультацій. Якщо студент з поважної причини пропустив захист лабораторної роботи, він може з дозволу

викладача захистити її під час консультації. Відсутність студента на захисті лабораторної роботи без поважної причини відповідає оцінці «0 балів», тобто призводить до заборгованості, яка повинна бути ліквідована студентом до початку підсумкового контролю з наступного модуля. Кінцевий термін ліквідації заборгованості з модульного контролю обмежується початком заліково-екзаменаційної сесії. Лабораторні роботи, які захищаються невчасно без поважних причин, оцінюються на нижчу оцінку (від -10% до -50% від максимальної кількості балів – залежно від терміну затримки здачі). Перезахист лабораторних робіт не дозволяється. Складання (перескладання) заліку відбувається за встановленим деканатом розкладом.

КРИТЕРІЇ ОЦІНЮВАННЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ

Оцінювання знань студентів здійснюється на основі результатів поточного, модульного та підсумкового контролю. Оцінювання здійснюється за програмним матеріалом навчальної дисципліни, засвоєння якого перевіряється пропонованими видами контролю.

Поточний контроль здійснюється під час проведення лабораторних занять і перевірки самостійної роботи студентів, а також під час читання лекцій. Модульний контроль здійснюється за результатами виконаних модульних контрольних робіт та перевірки лабораторних робіт і проектів. Завданнями поточного та модульного контролю є перевірка рівня розуміння та засвоєння лекційного матеріалу, набуття практичних навичок і досвіду виконання індивідуальних і комплексних задач.

Завданням підсумкового контролю (заліку) є перевірка розуміння студентом програмного матеріалу в цілому, здатності успішно розв'язувати поставлені практичні задачі та комплексно використовувати отримані знання.

Оцінювання знань здійснюється за 100-бальною шкалою. Результати роботи впродовж навчального семестру оцінюються в ході поточного та модульного контролю на інтервалі оцінок від 0 до 60 балів, а результати підсумкового контролю (заліку) оцінюються максимум у 40 балів. Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів поточного та модульного контролю і підсумкового контролю.

Розподіл балів з навчальної дисципліни

Поточне тестування та самостійна робота							Іспит	Сума
Змістовий модуль №1		Змістовий модуль №2			Змістовий модуль №3			
T1	T2	T1	T2	T3	T1	T2	40	100
10	5	10	10	10	10	5		

Іспит (40 балів) складається з онлайн-тестування на 20 балів та комплексного практичного завдання на 20 балів. Оцінювання результатів проходження онлайн-тестування здійснюється автоматично, на основі кількості наданих правильних відповідей на питання тесту. За суттєві помилки в розв'язуванні практичного завдання знімається 10-14 балів. За несуттєві помилки в розв'язуванні практичного завдання знімається 2-8 балів. Часткове розв'язання практичного завдання із грубими помилками оцінюється не вище 6 балів.

Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів, набраних студентом, згідно з наступною таблицею:

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
Зараховано	A (90-100)	відмінно
	B (80-89)	дуже добре
	C (70-79)	добре
	D (60-69)	задовільно
	E (50-59)	достатньо
Не зараховано	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (0-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компонента до перескладання

ПЕРЕЛІК ПИТАНЬ ДЛЯ САМОКОНТРОЛЮ ТА ПІДСУМКОВОГО КОНТРОЛЮ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ

1. Що таке GPU-архітектура та чим вона відрізняється від архітектури CPU?
2. Поясніть поняття масивного паралелізму та його реалізацію на графічних процесорах.
3. У чому полягає суть гетерогенного програмування GPGPU?
4. Які основні компоненти апаратної архітектури GPU?
5. Опишіть модель виконання програм у середовищі GPGPU.
6. Що таке технологія CUDA та які її основні переваги?
7. Опишіть архітектуру CUDA та модель програмування CUDA.
8. Що таке CUDA-ядро та як відбувається його виконання на GPU?
9. Які типи пам'яті використовуються в CUDA та для чого вони призначені?
10. Поясніть принципи управління пам'яттю в CUDA-програмах.
11. Що таке OpenCL та яку роль воно відіграє в гетерогенному програмуванні?
12. Опишіть модель платформи та виконання програм в OpenCL.
13. Як відбувається написання та запуск OpenCL-ядер?
14. У чому полягають особливості гетерогенних обчислень із використанням CPU та GPU в OpenCL?
15. Порівняйте технології CUDA та OpenCL: основні відмінності та сфери застосування.

16. Які основні напрямки застосування гетерогенного програмування та GPGPU в сучасних обчислювальних системах?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Коцовський В. М. К75 Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с.
2. CUDA by Example. An Introduction to General-Purpose GPU Programming. Addison-Wesley - Boston. July 2010.
3. R. Hochberg. Matrix Multiplication with CUDA. A basic introduction to the CUDA programming model. Cambridge University Press, August 11, 2012.
4. David B. Kirk, Wen-mei Hwu. Programming Massively Parallel Processors. A Hands-on Approach. - Published by Elsevier Inc. - 2010.
5. El Zein, A.H.; Rendell, A.P., Generating optimal CUDA sparse matrix–vector product implementations for evolving GPU hardware, Concurrency and Computation: Practice and Experience (2011).
6. B. Wilkinson and M. Allen. Parallel Programming Techniques & Applications Using Networked Workstations & Parallel Computers, 2nd ed. Toronto, Canada: Pearson, 2004.
7. Sun Microsystems. Multithreading Programming Guide. Santa Clara, CA: Sun Microsystems, 2008.

Додаткова (допоміжна)

1. Качко О.Г. Паралельне програмування. – Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2016. – 403 с.
2. Williams, S. W., D. Patterson, et. al. “The Roofline Model: A pedagogical tool for auto-tuning kernels on multicore architectures.” In Hot Chips, A Symposium on High Performance Chips, Vol. HC20 (August 10, 2008).
3. A. Grama, A. Gupta, G. Karypis, and V. Kumar. Introduction to Parallel Computing, 2nd ed. Reading, MA: Addison Wesley, 2003.
4. Suchard, M.A.; Wanq, Q.; Chan, C.; Frelinger, J.; Cron, A.; West, M., Understanding GPU Programming for Statistical Computation: Studies in Massively Parallel Massive Mixtures, J Comput Graph Stat. 19 (2) (2010, Jun 1) 419–438.
5. Micikevicius, P., 3D Finite Difference Computation on GPUs using CUDA. Proceeding GPGPU-2 Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units. (2010) ACM, New York, NY.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Електронний курс в системі Moodle
<https://moodle.chnu.edu.ua/course/view.php?id=9106>

2. Технологія CUDA

<https://developer.nvidia.com/cuda>

3. Технологія OpenCL

<https://www.khronos.org/opencl>

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича»

<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu-imeni-yurii-fedkovycha/>

- «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича»

<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yurii-fedkovycha/>

- «Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича»

<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yurii-fedkovycha/>

Використання здобувачами освіти інструментів штучного інтелекту (ШІ)

під час виконання навчальних завдань:

- допускається за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- не допускається використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- додаткові обмеження чи вимоги щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.