

Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

Дмитро ЧУМАЧЕНКО
(ім'я та ПРИЗВИЩЕ)

« 21 » 08 2025 р.

**СИЛАБУС ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Паралельні та розподілені обчислювання
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 122 «Комп'ютерні науки»
(код і найменування спеціальності)

Освітня програма: «Інтелектуальні системи та технології»
(найменування освітньої програми)

Рівень вищої освіти: *перший (бакалаврський)*

Силабус уведено в дію з 01.09.2025

Харків – 2025 р.

Розробник: Скоб Ю.О., проф. каф. 304, д. т. н., проф.
(прізвище та ініціали, посада, науковий ступінь і вчене звання)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри № 304
математичного моделювання та штучного інтелекту
(назва кафедри)

Протокол № 1 від « 21 » серпня 2025 р.

В. о. завідувача кафедри к.ф.-м.н., доцент
(науковий ступінь і вчене звання)


(підпис)

Олексій КАРТАШОВ
(ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:

Здобувач першого (бакалаврського)

рівня вищої освіти за ОП

«Інтелектуальні системи та технології»


(підпис)

Катерина ПИЛИПЕНКО
(ім'я та ПРІЗВИЩЕ)

1. Загальна інформація про викладача



ПІБ: *Скоб Юрій Олексійович*

Посада: *професор каф. 304*

Науковий ступінь: *докт. техн. наук*

Вчене звання: *професор*

Перелік дисциплін, які викладає:

Методи обчислень; Об'єктно-орієнтоване програмування (мова C#); Паралельні та розподілені обчислювання; Розподілені інформаційно-аналітичні системи

Напрями наукових досліджень:

Математичне моделювання наслідків впливу техногенних аварій, пов'язаних з викидом небезпечних газів у атмосферу, на навколишнє середовище

Контактна інформація:

y.skob@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	7-й
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	4,5 кредита ЄКТС / 135 годин (64 аудиторних, з яких: лекції – 32, лабораторні – 32; СРЗ – 71);
Види навчальної діяльності	Лекції, лабораторні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	«Організація та обробка електронної інформації», «Основи програмування (мова C++)», «Теорія алгоритмів і математична логіка», «Методи обчислень», «Об'єктно-орієнтоване програмування (мова C#)», «Операційні системи», «Web- програмування (.Net Framework)», «Enterprise додатки (мова Java)», «Бази даних та інформаційні системи», «Системне програмування», «Проектування програмного забезпечення», «Інтелектуальний аналіз даних»

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – формування у здобувачів вищої освіти системних знань і практичних навичок розроблення, аналізу та оптимізації програмних систем, що реалізують паралельні й розподілені обчислювальні процеси. Дисципліна спрямована на оволодіння сучасними концепціями, моделями та технологіями паралельних і розподілених обчислень, методами ефективного використання обчислювальних ресурсів, а також на розвиток здатності проектувати, програмно реалізовувати та досліджувати інтелектуальні системи з високим рівнем масштабованості, продуктивності та надійності.

Завдання – поетапне ознайомлення здобувачів вищої освіти із теоретичними основами, архітектурами та моделями паралельних і розподілених обчислень, принципами взаємодії процесів і потоків, синхронізації та обміну даними. Дисципліна спрямована на формування практичних навичок розроблення, налагодження та оптимізації програм із використанням сучасних інструментів і технологій паралельного програмування, а також на розвиток здатності застосовувати розподілені обчислювальні моделі, хмарні платформи та мікросервісні архітектури для створення інтелектуальних інформаційних систем.

Компетентності, які набуваються:

Інтегральна компетентність:

здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде здатен:

- до абстрактного мислення, аналізу та синтезу (ЗК 1);
- демонструвати знання та розуміння предметної області та розуміння професійної діяльності (ЗК 3);
- спілкуватися державною мовою як усно, так і письмово (ЗК 4);
- спілкуватися іноземною мовою (ЗК 5).

Спеціальні компетентності (ФК)

Після закінчення цієї програми здобувач освіти буде здатен:

- до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і

підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування (СК 1);

- до виявлення статистичних закономірностей недетермінованих явищ, застосування методів обчислювального інтелекту, зокрема статистичної, нейромережевої та нечіткої обробки даних, методів машинного навчання та генетичного програмування тощо (СК 2);

- до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (СК 3);

- використовувати сучасні методи математичного моделювання об'єктів, процесів і явищ, розробляти моделі й алгоритми чисельного розв'язування задач математичного моделювання, враховувати похибки наближеного чисельного розв'язування професійних задач (СК 4);

- до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризики (СК 6);

- застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів (СК 7);

- реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах (СК 9);

- до інтелектуального аналізу даних на основі методів обчислювального інтелекту включно з великими та погано структурованими даними, їхньої оперативної обробки та візуалізації результатів аналізу в процесі розв'язування прикладних задач (СК 11);

- забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення (СК 12);

- реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації (СК 16);

– розробляти інтелектуальне програмне забезпечення, використовувати сучасні методи штучного інтелекту (СК 17).

Програмні результати навчання (ПРН):

– Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук (ПРН 1);

– Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проєктування та реалізації об'єктів інформатизації (ПРН 2);

– Використовувати знання закономірностей випадкових явищ, їх властивостей та операцій над ними, моделей випадкових процесів та сучасних програмних середовищ для розв'язування задач статистичної обробки даних і побудови прогнозних моделей (ПРН 3);

– Використовувати методи обчислювального інтелекту, машинного навчання, нейромережевої та нечіткої обробки даних, генетичного та еволюційного програмування для розв'язання задач розпізнавання, прогнозування, класифікації, ідентифікації об'єктів керування тощо (ПРН 4);

– Використовувати методи чисельного диференціювання та інтегрування функцій, розв'язання звичайних диференціальних та інтегральних рівнянь, особливостей чисельних методів та можливостей їх адаптації до інженерних задач, мати навички програмної реалізації чисельних методів (ПРН 6);

– Розуміти принципи моделювання організаційно-технічних систем і операцій, використовувати методи дослідження операцій, розв'язання одно- та багатокритеріальних оптимізаційних задач лінійного, цілочисельного, нелінійного, стохастичного програмування (ПРН 7);

– Розробляти моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук (ПРН 9);

– Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт) (ПРН 11);

– Застосовувати методи та алгоритми обчислювального інтелекту та інтелектуального аналізу даних в задачах класифікації, прогнозування, кластерного аналізу, пошуку асоціативних правил з використанням

програмних інструментів підтримки багатовимірного аналізу даних на основі технології Data Mining, Text Mining, Web Mining (ПРН 12);

– Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення (ПРН 16);

– Використовувати сучасні підходи та методи штучного інтелекту (ПРН 17).

4. Зміст навчальної дисципліни

МОДУЛЬ

Змістовний модуль 1. Програмування задач

Тема 1. Способи створення задач

Стисла анотація: Предмет вивчення і задачі дисципліни «Паралельні та розподілені обчислювання». Основні історичні етапи розвитку і становлення сучасної моделі проектування і програмування, перспективи подальшого розвитку. Паралельне програмування в .NET. Переваги та недоліки паралельного програмування. Облік накладних витрат. Координація даних. Масштабування додатків. Коли слід виконувати паралельно. Вирішуючи, коли виконувати послідовно. Ознайомлення з класом Task. Створення та запуск завдань. Створення простих завдань. Налаштування стану завдання. Як отримати результат. Завдання параметрів створення задачі. Ідентифікація задач.

Теми лекцій і практичних занять: «Способи створення задач».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, інсталяція інтегрованого середовища Visual Studio C# з офіційного сайту Microsoft, виконання індивідуальних завдань по створенню задач відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 2. Скасування задач

Стисла анотація: Скасування однієї задачі. Моніторинг скасування за допомогою опитування. Моніторинг скасування за допомогою делегата. Моніторинг скасування за допомогою дескриптора очікування. Скасування декількох задач. Створення композитних маркерів скасування. Визначення того, чи була задача скасована.

Теми лекцій і лабораторного заняття: «Скасування задач».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання індивідуального завдання з використанням скасування задач відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 3. Режим сну задач

Стисла анотація: В очікуванні часу продовження виконання задачі. Використання дескриптора очікування маркера скасування. Використання класичного очікування. Використання спінового очікування. Очікування

задач. В очікуванні єдиної задачі. Очікування декількох задач. В очікуванні однією з багатьох задач. Ліниве виконання задач.

Теми лекцій і практичних занять: «Режим сну задач».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням режиму сну задач відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 4. Оброблення виключень в задачах

Стисла анотація: Загальні проблеми під час виконання паралельних задач та їх причини. Тупик залежностей задач. Оцінений конфлікт локальних змінних. Надмірний спінінг. Механізми оброблення виключних ситуацій у паралельних та асинхронних обчисленнях мовою C#. Принципи перехоплення та поширення виключень у задачах (Task), об'єднаних задачах (Task.WhenAll, Task.WhenAny), методи діагностики й оброблення помилок у багатопоточних і розподілених програмах. Використання типу AggregateException, засобів логування та забезпеченню надійності виконання паралельних обчислювальних процесів.

Теми лекції і лабораторного заняття: «Оброблення виключень в задачах».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуального завдання з використанням оброблення виключень в задачах відповідно умов варіанту, підготовка до захисту лабораторної роботи, підготовка до модульного контролю.

Модульний контроль 1

Змістовний модуль 2. Спільне використання даних

Тема 1. Спільне використання даних

Стисла анотація: Проблеми, що виникають під час одночасного доступу кількох потоків до спільних змінних або структур даних у паралельних програмах. Аналіз типових помилок, пов'язаних з неконтрольованою модифікацією спільних ресурсів, включно з гонкою за даними (race condition), неузгодженістю станів та непередбачуваністю результатів обчислень. Підходи до впорядкування виконання операцій над даними – створення детермінованого порядку виконання, послідовна та атомарна обробка, а також принципи виконання без зміни стану (immutability) і виконання в ізоляції (isolation) як засоби запобігання конфліктам доступу. Стратегії організації безпечного спільного використання даних у C#: використання незмінних типів, копіювання при записі (copy-on-write), блокування доступу до об'єктів, локальні копії змінних потоку (ThreadLocal), а також застосування колекцій з підтримкою потокобезпеки з простору імен System.Collections.Concurrent.

Теми лекцій і практичних занять: «Спільне використання даних».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням спільного використання даних відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 2. Синхронізація використання даних

Стисла анотація: Принципи і механізми синхронізації доступу до спільних ресурсів у паралельних і розподілених обчисленнях. Поняття критичних ділянок коду, типи конфліктів при одночасному доступі до даних, причини виникнення станів гонитви (race conditions) та методи їх запобігання. Основні примітиви синхронізації, що реалізовані у платформі .NET: lock, Monitor, Mutex, Semaphore, ReaderWriterLockSlim, SpinLock, Barrier, CountdownEvent тощо. Аналіз принципів їх роботи, сфери доцільного застосування, а також типові помилки при синхронізації потоків. Оптимальне використання синхронізації для досягнення балансу між безпечністю доступу до даних і продуктивністю виконання програм. Практичні стратегії уникнення блокувань (deadlocks), зниження конкуренції між потоками та забезпечення узгодженості даних у багатопоточних і асинхронних системах.

Теми лекцій і практичних занять: «Синхронізація використання даних».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням синхронізації використання даних відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 3. Блокування спільних даних

Стисла анотація: Блокування та моніторинг. Використання заблокованих операцій. Використання спінової блокування. Використання дескрипторів очікування і класу Mutex. Конфігурування міжпроцесної синхронізації. Використання декларативної синхронізації. Використання блокувань доступу «читання-запис». Неочікувана мінливість. Множинні блокування. Порядок отримання блокування. Осиротілі блокування.

Теми лекцій і практичних занять: «Блокування спільних даних».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням блокування спільних даних відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 4. Класи спільних колекцій

Стисла анотація: Спеціалізовані колекції .NET, призначені для безпечного спільного доступу з кількох потоків у паралельних і розподілених програмах. Класи простору імен System.Collections.Concurrent, зокрема ConcurrentDictionary, ConcurrentQueue, ConcurrentBag, BlockingCollection, їх структура, принципи роботи та відмінності від звичайних колекцій. Аналіз практичних прийомів забезпечення потокобезпеки, ефективності та узгодженості даних у багатопоточних застосунках.

Теми лекцій і практичних занять: «Класи спільних колекцій».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням класів спільних колекцій

відповідно умов варіанту, підготовка до захисту лабораторної роботи, підготовка до модульного контролю.

Модульний контроль 2

Змістовний модуль 3. Координація задач

Тема 1. Продовження задач

Стисла анотація: Механізми побудови та керування ланцюгами залежних задач (продовжень, continuations) у паралельному програмуванні мовою C#. Принципи створення продовжень на основі об'єктів класу Task, а також застосування методів ContinueWith, Task.WhenAll, Task.WhenAny та асинхронних делегатів для побудови складних обчислювальних процесів. Різні моделі зв'язків між задачами: просте продовження (одна задача запускає наступну після завершення), схеми “один-до-багатьох”, “багато-до-одного” та “будь-який-до-одного”. Практичні аспекти створення вибіркового продовжень, які виконуються залежно від стану попередньої задачі – успішного завершення, скасування чи виникнення виключення. Ефективне використання продовжень для керування асинхронними потоками виконання, підвищення продуктивності та покращення читабельності коду у складних паралельних застосунках. Приклади комбінування продовжень з механізмами синхронізації, скасування та оброблення виключень, що дозволяють створювати стійкі й масштабовані обчислювальні системи.

Теми лекцій і практичних занять: «Продовження задач».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням продовження задач відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 2. Скасування та виключення продовжень

Стисла анотація: Механізми керування життєвим циклом паралельних задач та їхніх продовжень (continuations) у середовищі .NET. Принципи скасування задач за допомогою токенів CancellationToken, координації виконання кількох задач і безпечного завершення обчислень у випадках відмов чи перевантаження системи. Способи очікування та синхронізації продовжень за допомогою методів Task.ContinueWith, Task.WhenAll, Task.WhenAny, а також особливості побудови асинхронних ланцюжків задач, у яких результат однієї операції впливає на виконання наступних. Оброблення виключень у продовженнях: механізми перехоплення та агрегації помилок через AggregateException, особливості передачі виключень між потоками, вплив виключних ситуацій на подальше виконання задач і забезпечення узгодженості станів при помилках.

Теми лекцій і практичних занять: «Скасування та виключення продовжень».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням скасування та обробки виключень продовжень відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 3. Синхронізація задач

Стисла анотація: Принципи координації виконання паралельних задач у середовищі .NET із використанням механізмів синхронізації. Забезпечення узгодженого виконання потоків, уникнення конфліктів доступу до спільних ресурсів і побудова надійної архітектури взаємодії задач у паралельних програмах. Створення дочірніх задач (child tasks) і принципи їхньої участі в життєвому циклі батьківських задач. Розглядається поведінка задач при використанні опцій вкладеності та приєднання (AttachedToParent), що дозволяє контролювати завершення та оброблення помилок у складних ієрархіях паралельного виконання. Синхронізаційні примітиви .NET, які забезпечують ефективну координацію між задачами: бар'єри (Barrier) – для узгодження етапів виконання кількох потоків; CountdownEvent – для очікування завершення певної кількості операцій; ManualResetEventSlim і AutoResetEvent – для сигналізації між потоками; SemaphoreSlim – для обмеження одночасного доступу до ресурсів. Застосування цих механізмів у практичних шаблонах паралельного програмування, зокрема у шаблоні «виробник–споживач (producer–consumer pattern)», який дозволяє організувати асинхронну обробку потоків даних. Методи побудови цього шаблону із використанням черг блокування (BlockingCollection), об'єднання множин колекцій та управління взаємодією між кількома джерелами даних. Створення користувацьких планувальників задач (custom task schedulers), що дає змогу реалізувати специфічні політики планування – наприклад, пріоритетне виконання задач, обмеження паралелізму або контроль за розподілом ресурсів.

Теми лекцій і практичних занять: «Синхронізація задач».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням синхронізації задач відповідно умов варіанту, підготовка до захисту лабораторної роботи, підготовка до модульного контролю.

Модульний контроль 3

Змістовний модуль 4. Паралельні цикли

Тема 1. Паралельні цикли

Стисла анотація: Принципи організації паралельного виконання ітерацій циклів у середовищі .NET із використанням класу Parallel. Відмінності між послідовними та паралельними циклами, вплив

розпаралелювання на продуктивність. Особливості планування та балансування навантаження між потоками. Основні методи паралельного виконання ітерацій – `Parallel.For` та `Parallel.ForEach`, їх синтаксис, параметри, способи передавання дій (`Action`, `Func`) до тіла циклу. Аналіз типових сценаріїв застосування паралельних циклів у задачах оброблення великих масивів даних, симуляцій, наукових і графічних обчислень. Налаштування параметрів виконання за допомогою класу `ParallelOptions`: обмеження кількості потоків (`MaxDegreeOfParallelism`), використання токенів скасування (`CancellationToken`), контроль завершення виконання (`ParallelLoopResult`). Підходи до синхронізації доступу до спільних змінних у межах циклу, уникнення конфліктів даних і забезпечення потокобезпечності.

Теми лекцій і практичних занять: «Паралельні цикли».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням паралельних циклів відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 2. Керування паралельними циклами

Стисла анотація: Методи контролю виконання паралельних циклів у середовищі .NET з використанням бібліотеки `Task Parallel Library (TPL)` і класу `Parallel`. Засоби переривання, зупинки та скасування виконання циклів, зокрема механізми дострокового завершення ітерацій за допомогою методів `Break()` та `Stop()`, а також використання токенів скасування (`CancellationToken`) для узгодженого завершення обчислень. Підходи до оброблення виключень у паралельних циклах, зокрема механізми агрегації помилок через `AggregateException`, порядок їх виявлення та вплив на подальше виконання програми. Отримування результатів виконання паралельних циклів за допомогою об'єкта `ParallelLoopResult` і аналіз стану обчислень. Використання потокової локальної пам'яті (`Thread Local Storage`) для зберігання проміжних результатів кожного потоку, що забезпечує незалежність виконання і підвищує продуктивність. Підходи до реалізації паралельних циклів із залежностями, коли порядок виконання ітерацій впливає на коректність результатів, та методи забезпечення узгодженості даних у таких випадках.

Теми лекцій і практичних занять: «Керування паралельними циклами».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням керування паралельними циклами відповідно умов варіанту, підготовка до захисту лабораторної роботи.

Тема 3. Розподільники паралельних циклів

Стисла анотація: Принципи і стратегії розподілу роботи між потоками у паралельних циклах, що реалізуються за допомогою бібліотеки `Task Parallel`

Library (TPL) у середовищі .NET. Механізми керування розподілом ітерацій між потоками для досягнення оптимального завантаження процесорів і мінімізації накладних витрат синхронізації. Основи вибору стратегії розбиття задачі – від статичного до динамічного розподілу. Аналіз chunking-стратегії («розбиття на шматки»), за якої ітерації групуються в блоки (chunks), що призначаються потокам для виконання. Пояснюється, як розмір блоку впливає на баланс між продуктивністю та рівномірністю завантаження процесорів. Впорядкований розподіл за замовчуванням, який забезпечує передбачуваність послідовності оброблення даних, а також механізми його налаштування в залежності від характеру задачі. Розгляд ситуацій, коли доцільно використовувати невпорядкований розподіл для підвищення швидкодії. Методи створення користувацьких розподільників (custom partitioners), що дозволяють адаптувати процес розподілу під специфіку конкретних обчислень, наприклад при роботі з нестандартними структурами даних або при необхідності динамічного балансування навантаження. Написання контекстного розподільника (contextual partitioner), який враховує контекст виконання – наприклад, залежності між даними або пріоритети задач.

Теми лекції і практичних занять: «Розподільники паралельних циклів».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань з використанням розподільників паралельних циклів відповідно умов варіанту, підготовка до захисту контрольної роботи, , підготовка до модульного контролю.

Модульний контроль 4

5. Індивідуальні завдання – немає.

6. Методи навчання

1. Пояснювально-ілюстративний (інформаційно-рецептивний) метод: та метод проблемного виконання (лекція).
2. Репродуктивний (лабораторні роботи).
3. Частково-пошуковий (евристичний) та дослідницький: (самостійна робота).
4. Дисципліна «Паралельні та розподілені обчислювання» передбачає лекційні (у т. ч. з використанням мультимедійного обладнання) і лабораторні заняття під керівництвом викладача та самостійну роботу студента за підручниками і матеріалами, опублікованими кафедрою (методичні посібники та мережеві ресурси), що забезпечує закріплення теоретичних знань, сприяє набуттю практичних навичок і розвитку самостійного наукового мислення. Передбачено регулярні індивідуальні консультації.

7. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних (захист лабораторних робіт) і підсумкових контролів (захист змістовного модуля, іспит).

У завдання поточного контролю входить систематична перевірка розуміння й засвоєння студентом програмного матеріалу, виконання лабораторних робіт, умінь самостійно опрацьовувати тексти програм розрахунку, написання звітів, здатності усно або письмово представляти певний матеріал дослідження.

Перед підсумковим контролем ставиться завдання перевірки глибини засвоєння студентом програмного матеріалу дисципліни, логіки й взаємозв'язки між її окремими розділами, здатності творчо використовувати набуті знання, умінь сформулювати своє відношення до проблеми, що впливає зі змісту навчальної дисципліни.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Виконання і захист лабораторних робіт	0...5	4	0...20
Модульний контроль	0...5	1	0...5
Змістовний модуль 2			
Виконання і захист лабораторних робіт	0...5	4	0...20
Модульний контроль	0...5	1	0...5
Змістовний модуль 3			
Виконання і захист лабораторних робіт	0...5	3	0...15
Модульний контроль	0...10	1	0...10
Змістовний модуль 4			
Виконання і захист лабораторних робіт	0...5	3	0...15
Модульний контроль	0...10	1	0...10
Усього за семестр			0...100

Семестровий контроль (*іспит*) проводиться у разі відмови здобувача освіти від балів підсумкового контролю й за наявності допуску до *іспиту*. Під час складання семестрового *іспиту* здобувач освіти має можливість отримати максимум 100 балів.

Білет для *іспиту* складається з одного теоретичного і двох практичних питань. У першому запитанні студент має продемонструвати теоретичні

знання. У другому запитанні – показати навички складання і виконання програми з використанням об’єктів задач. У третьому пункті – продемонструвати знання зі створення і застосування об’єктів паралельного доступу до спільних даних мовою C# в інтегрованому середовищі Visual Studio на комп’ютері з багатоядерним процесором. Максимальна кількість балів за кожне запитання та їх суми показані в табл. 8.2.

Таблиця 8.2 – Шкали оцінювання відповідей на запитання білету

Складові білету	Складові оцінки	Бали за одне питання	Сумарна кількість балів
Пункт 1. Теоретичне запитання	Надано визначення поняття	0...5	0...20
	Наведено приклад використання	0...5	
	Наведено фрагмент коду	0...10	
Пункт 2. Практичне запитання	Складено алгоритм розрахунку (блок-схема)	0...10	0...30
	Створено проект консольного додатку з використанням задач	0...10	
	Отримано правильні результати у консольному вікні	0...10	
Пункт 3. Практичне запитання	Складено блок-схему розрахунку	0...15	0...50
	Створено проект застосунку з паралельним доступом до спільних даних	0...20	
	Отримано правильні результати у терміналі або формі	0...15	
Підсумкова оцінка за іспит			0...100

Таблиця 8.3 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Задовільно (60-74) – мати знання і уміння для забезпечення програмних результатів навчання. Виконати та здати лабораторні роботи. Написати модульні роботи. Уміти створювати проекти консольного типу в інтегрованому середовищі розробки програмного забезпечення Visual Studio мовою C# на комп’ютері з багатоядерним процесором для проведення

паралельних обчислень з використанням класу задач. Знати основи реалізації типових завдань паралельної обробки інформації.

Добре (75-89) – мати знання і уміння для забезпечення програмних результатів навчання. Виконати та здати лабораторні роботи. Написати модульні роботи. Додатково до вимог, які визначено для отримання задовільної оцінки: уміти створювати алгоритм паралельного розрахунку у вигляді блок-схеми; уміти виконувати завдання у вигляді консольного застосунку але з використанням примітивів синхронізації обчислювальних потоків; уміти реалізовувати виконання варіантів специфічних умов завдань, які відрізняються від типових.

Відмінно (90-100) – мати знання, уміння й навички, що дадуть змогу самостійно, вільно та обґрунтовано відповідати на будь-які питання щодо методології та реалізації необхідних паралельних розрахунків мовою C# на комп'ютері з багатоядерним процесором. Досконало виконати та успішно здати лабораторні роботи. Успішно написати модульні роботи. Уміти самостійно використовувати інтегроване середовище Visual Studio для розробки та відлагодження додатку мовою C# для паралельної обробки інформації. Досконало вміти використовувати інструменти такі інструменти паралельної обробки даних, як спільне використання даних, синхронізація використання даних, блокування спільних даних, класи спільних колекцій, продовження задач, скасування та виключення продовжень, синхронізація задач, паралельні цикли.

9. Політика навчального курсу

Відвідування занять. Регуляція пропусків. Інтерактивний характер курсу передбачає обов'язкове відвідування практичних занять. Здобувачі освіти, які за певних обставин не можуть відвідувати лабораторні заняття регулярно, мають протягом тижня узгодити із викладачем графік індивідуального відпрацювання пропущених занять. Окремі пропущені заняття мають бути відпрацьовані на найближчій консультації протягом тижня після їх пропуску. Відпрацювання занять здійснюється усно у формі співбесіди за питаннями, визначеними планом заняття. В окремих випадках дозволяється письмове відпрацювання пропущених занять шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується,

що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикавання джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu.ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

1. Паралельні та розподілені обчислення / Ю.О. Скоб, В. О. Халтурін. – Навч. посібник до лаб. практик. – Харків: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2023. – 116 с. Режим доступу: <http://library.khai.edu/library/fulltexts/metod/Paralelni-Ta-Rozpodileni-Obchyslennya.pdf>.

2. Object oriented programming using C# / Y.O. Skob, V. O. Khalturin. – Laboratory course study guide. – Kharkiv : KhAI, 2020. – 109 p. Режим доступу: http://library.khai.edu/library/fulltexts/doc/object_oriented_programming_using.pdf.

3. Основи програмування мовою C# [Електронний ресурс] : навч. посіб. до лаб. практик. / Ю. О. Скоб, М. Л. Угрюмов, В. О. Халтурін. – Х. : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харьк. авіац. ін-т», 2017. – 108 с. Режим доступу: http://library.khai.edu/library/fulltexts/metod/Skob_Osnovi_Programuvannya_movoyu_C#.pdf.

4. Programming and numerical methods. Part 2: Numerical methods using MatLab and Mathcad : guidance manual for laboratory works / O. V. Yarova, D. I. Chumachenko. – Kharkiv : National Aerospace University «KhAI», 2016. – 93 p. Режим доступу: http://library.khai.edu/library/fulltexts/metod/Yarova_Program_Method.pdf.

5. Основи програмування мовою C++. Вступ до ООП / К. П. Коробчинський, І. В. Москович, Ю.О. Скоб, О. С. Пічугіна. – Навч. посібник до лаб. практик. – Харків: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2024. – 124 с. Режим доступу: http://library.khai.edu/library/fulltexts/metod/Osnovy_Programuvannya_Movoyu_C++.pdf.

6. Основи програмування сучасним Фортраном / Ю.О. Скоб, М. Л. Угрюмов, В. О. Халтурін. – Навч. посібник до лаб. практик. – Харків: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2016. – 96 с. Режим доступу: http://library.khai.edu/library/fulltexts/metod/Skob_Osnovi_Programuvanna.pdf.

Посилання на курс у системі дистанційного навчання Ментор:
<https://mentor.khai.edu/course/view.php?id=3148>.

11. Рекомендована література

Базова

1. Минайленко Р. М. Паралельні та розподілені обчислення : навч. посіб. / Р. М. Минайленко. – Кропивницький : Центральнотукаїнський нац. техн. ун-т, 2021. – 120 с. – Режим доступу: <https://dspace.kntu.kr.ua/server/api/core/bitstreams/396e02d2-725b-47b5-a1c0-ae07a9bec326/content>
2. Матвеева Н. О., Пономарьов І. В., Пляка С. М. Паралельне програмування на платформі .NET : навч. посіб. / Н. О. Матвеева, І. В. Пономарьов, С. М. Пляка. – Дніпро : Дніпровський нац. ун-т, 2023. – 210 с.
3. Семеренко С. І. Технології паралельних обчислень : навч. посіб. / С. І. Семеренко. – Вінниця : ВНТУ, 2018. – 104 с. – Режим доступу: https://pdf.lib.vntu.edu.ua/books/IRVC/2021/Semerenco_2018_104.pdf
4. Коцовський В. М. Технології розподілених систем та паралельних обчислень : метод. матеріали / В. М. Коцовський. – Ужгород : УжНУ, 2020. – 72 с. – Режим доступу: <https://dspace.uzhnu.edu.ua/server/api/core/bitstreams/8e13d04f-22ba-425a-a62a-c2ccce5ea9af/content>
5. Паралельне програмування : метод. матеріали до виконання лабораторних робіт / ІПЗЕ КПІ ім. І. Сікорського. – Київ, 2024. – 65 с. – Режим доступу: https://ipze.kpi.ua/wp-content/uploads/2024/01/ПВ-5-сем-Паралельне-програмування_Денна.pdf
6. Закон України «Про вищу освіту» від 01.07.2014 р. № 1556-VII // *Відомості Верховної Ради України*. – 2014. – № 37–38. – Ст. 2004. Режим доступу: <https://zakon.rada.gov.ua/laws/show/1556-18>
7. Постанова Кабінету Міністрів України «Про затвердження Положення про акредитацію освітніх програм, за якими здійснюється підготовка здобувачів вищої освіти» від 29.12.2021 р. № 1324. Режим доступу: <https://zakon.rada.gov.ua/go/z1608-21>
8. Міністерство освіти і науки України. *Методичні рекомендації щодо розроблення стандартів вищої освіти*. – Наказ МОН України від 01.06.2024 р. № 272. Режим доступу: <https://zakon.rada.gov.ua/go/v0022729-24>
9. Національне агентство із забезпечення якості вищої освіти. *Роз'яснення щодо застосування Критеріїв оцінювання якості освітньої програми*. – Київ, 2024. Режим доступу: <https://naqa.gov.ua>
10. Міністерство освіти і науки України. *Затверджені стандарти вищої освіти: офіційний перелік*. – Київ: МОН, 2024. Режим доступу: <https://mon.gov.ua/osvita-2/vishcha-osvita-ta-osvita-doroslikh/naukovo-metodichna-rada-ministerstva-osviti-i-nauki-ukraini/zatverdzeni-standarti-vishchoi-osviti>
11. Державна служба якості освіти України. *Методичні рекомендації щодо розроблення освітньо-професійних програм фахової передвищої освіти*.

– Київ, 2022. Режим доступу: https://sqe.gov.ua/wp-content/uploads/2022/06/Metodichni_rekomendacii_rozroblennya_OOP_FPO_2022.pdf

12. Національна академія наук України, Придніпровська державна академія будівництва та архітектури. *Проект стандарту освітньо-професійної програми (ОП-03-2020)*. – Дніпро, 2020. Режим доступу: <https://pgasa.dp.ua/wp-content/uploads/2020/04/proekt-standartu-OP.pdf>

Допоміжна

1. Коноваленко І. В. Платформа .NET та мова програмування C# 8.0 : навч. посіб. / І. В. Коноваленко. – Тернопіль : ТНТУ, 2020. – 180 с. – Режим доступу: <https://elartu.tntu.edu.ua/bitstream/lib/32825/1/Konovalenko%20I.%20.NET-C%23.pdf>

2. Нестерко А. Б. Обчислювальна техніка та програмування : практикум / А. Б. Нестерко. – Київ : КПІ ім. І. Сікорського, 2022. – 86 с. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/48838/1/OT_ta_P2_Prak.pdf

3. Смірнов В. В., Смірнова Н. В. Паралельні та розподілені обчислення : метод. вказівки до виконання самост. робіт / В. В. Смірнов, Н. В. Смірнова. – Кропивницький : ЦНТУ, 2022. – 58 с. – Режим доступу: <https://dspace.kntu.kr.ua/bitstreams/1c59a6d8-f4d1-42b2-be5d-669885a67b4d/download>

4. Метелап В. В. Лабораторна робота: паралельне програмування та бібліотека TPL [Електронний ресурс] / В. В. Метелап. – 2024. – Режим доступу: https://er.chdtu.edu.ua/bitstream/ChSTU/5610/1/MP_LP_ПРОГРАМУВАННЯ%20НА.NET_2024.pdf

5. Матвєєва Н. О. Паралельне програмування на платформі .NET : навч. посіб. [Електронний ресурс] / Н. О. Матвєєва, та ін. – ResearchGate, 2023. – Режим доступу: https://www.researchgate.net/publication/371178942_PARALELNE_PROGRAMUVANNA_NA_PLATFORMI_NET.

6. Семеренко В. П. Технології паралельних обчислень : навч. посіб. [Електронний ресурс] / В. П. Семеренко. – Вінниця : ВНТУ, 2018. – Режим доступу: https://pdf.lib.vntu.edu.ua/books/IRVC/2021/Semerenko_2018_104.pdf.

7. Купін А. І., Музика І. О. Мережні інформаційні технології. Практикум : навч. посіб. [Електронний ресурс] / А. І. Купін, І. О. Музика. – Кривий Ріг, 2015. – Режим доступу: https://lib.iitta.gov.ua/id/eprint/706684/1/Kupin_Muzyka_8.pdf.

8. Рольщиков В. Б. Distributed Systems Technology and Parallel Computing : Lecture Notes / Курси лекцій з паралельних і розподілених систем [Електронний ресурс] / В. Б. Рольщиков. – 2018. – Режим доступу: https://eprints.library.odeku.edu.ua/695/1/RolshchikovVB_Distributed_Systems_Technology_And_Parallel_Computing_Module_1_KL_2018.pdf.

12. Інформаційні ресурси

1. Microsoft Learn. Parallel programming in .NET [Електронний ресурс]. – Офіційна документація Microsoft, 2025. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/standard/parallel-programming/>.

2. Joseph Albahari. Threading in C# [Електронний ресурс] / Albahari.com, 2024. – Режим доступу: <https://www.albahari.com/threading/>.
3. Microsoft Learn. Distributed applications and microservices in .NET [Електронний ресурс]. – Офіційна документація Microsoft, 2025. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/>.
4. Harrison Ferrone. *Learning C# by Developing Games with Unity 2023 : practical introduction to C# and parallel programming*. – Packt Publishing, 2023. – 430 p.
5. Joseph D., Skeel R. *An Introduction to Parallel Computing : Design and Analysis of Algorithms*. – McGraw-Hill Education, 2021. – 512 p.