



ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ТА РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>4кред/120год (лекцій 36год., комп.прк.18 год., 66 СРС)</i>
Семестровий контроль/ контрольні заходи	<i>Залік/МКР</i>
Розклад занять	<i>http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., Варава Іван Андрійович, varava_ivan@ill.kpi.ua, @Varavalvan (у робочий час)</i> <i>Комп'ютерні практикуми: к.т.н., Варава Іван Андрійович, varava_ivan@ill.kpi.ua, @Varavalvan (у робочий час)</i>
Розміщення курсу	<i>Кампус</i>

Програма навчальної дисципліни

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

Силабус навчальної дисципліни «Технології паралельних та розподілених обчислень» складено відповідно до освітньої програми підготовки бакалаврів «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 – Інженерія програмного забезпечення.

Метою навчальної дисципліни є набуття студентами теоретичних знань та практичних навичок розробки додатків для паралельних та розподілених комп'ютерних систем, формування та закріплення наступних компетентностей:

(ЗК 1) Здатність до абстрактного мислення, аналізу та синтезу. (ЗК 2) Здатність застосовувати знання у практичних ситуаціях.

(ФК 3) Здатність розробляти архітектури, модулі та компоненти програмних систем. (ФК 7) Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. (ФК 14) Здатність до алгоритмічного та логічного мислення.

Предмет навчальної дисципліни – сучасні програмні пакети, сервіси, бібліотеки для візуалізації даних.

Програмні результати навчання, на формування та покращення яких спрямована дисципліна: (ПРН 6) Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; (ПРН 12) Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; (ПРН 13) Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. (ПРН 18) Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі результати навчання:

ЗНАННЯ:

- принципи розпаралелювання обчислювальних алгоритмів;
- методи програмування потоків і їх синхронізації;
- функціональний підхід в паралельних обчисленнях;
- архітектурні шаблони розподілених систем.

УМІННЯ:

- розробляти багатопоточні додатки для паралельних обчислень;
- аналізувати ефективність паралельних алгоритмів;
- організовувати міжпотоківу взаємодію;
- розробляти розподілені програмні системи.

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

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Основи програмування», «Алгоритми та структури даних», «Основи комп'ютерних мереж і систем».

Постреквізити дисципліни. Компетенції, знання та уміння, одержані в процесі вивчення кредитного модуля «Технології паралельних та розподілених обчислень» можуть бути використані при вивченні дисциплін «Побудова масштабованих систем обробки даних у реальному часі», «Переддипломна практика» та «Дипломне проектування».

3. Зміст навчальної дисципліни

Розділ 1. Основи паралельних обчислень

Тема 1.1 Основи паралельних обчислювальних систем

Тема 1.2 Засоби синхронізації паралельних програм

Тема 1.3 Прикладні програмні інтерфейси для паралельних обчислень

Розділ 2. Засоби розподілених обчислень в Java

Тема 2.1 Конкурентність в Java

Тема 2.2 Функціональне програмування в Java

Тема 2.3 Реактивне програмування в Java

Розділ 3. Технології розподілених систем

Тема 3.1 Архітектурні шаблони розподілених систем

Тема 3.2 Робота з даними в розподілених системах

Тема 3.3 Побудова відмовостійких розподілених систем

4. Навчальні матеріали та ресурси

Основна література

1. Технології паралельних обчислень : навчальний посібник / Семеренко В. П. – Вінниця : ВНТУ, 2018. – 104 с.
2. Robey R. Zamora Y. *Parallel and High Computing* / Manning. – 2021. – 704 p.
3. Thomas Rauber, Gudula Rünger *Parallel Programming*, 2nd Ed. Springer-Verlag Berlin Heidelberg, 2013. – 516 p.
4. Naftalin, Maurice *Mastering lambdas: Java programming in a multicore world* McGraw-Hill Education, 2015. – 208 p.
5. Gavin M. Roy *RabbitMQ in Depth* / Manning. – 2017. – 264p.

Додаткова література

6. Home – OpenMP [Електронний ресурс]. – Режим доступу : <https://www.openmp.org/>. – Назва з екрану.
7. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft *Modern Java in Action: Lambdas, streams, functional and reactive programming* / Manning Publications, 2018. – 592 p.
8. Atomix [Електронний ресурс]. – Режим доступу : <https://atomix.io/>. – Назва з екрану.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела)
1	Тема 1.1 Паралельні обчислювальні системи. <u>Основні питання:</u> Класифікація паралельних обчислювальних систем. Розпаралелювання програм.
2	Тема 1.1 Вступ до паралельних обчислень. <u>Основні питання:</u> Паралельні обчислення. Типи паралелізму. Оцінка ефективності паралельних алгоритмів. Закон Амдала. Закон Густавсона-Барсіса.
3	Тема 1.2. Потoki. <u>Основні питання:</u> Створення потоків. Синхронізація потоків. Принципи синхронізації. Атомарність. Методи синхронізації. Семафори. Монітори.
4	Тема 1.2 Класичні задачі синхронізації <u>Основні питання:</u> Задача про обідаючих філософів. Задача «читачів та письменників». Задача «виробників та споживачів». Задача про сплячого перукаря. Дедлоки. Stateless- та Immutable-об'єкти.
5	Тема 1.2 Неблокуючі алгоритми <u>Основні питання:</u> Три рівні неблокуючої синхронізації. Атомарні операції. Операція атомарної зміни(CAS). ABA-проблема. Алгоритм Майкла-Скотта.
6	Тема 1.3. Технології OpenMP та MPI <u>Основні питання:</u> Програмна модель OpenMP. Директиви та функції OpenMP. Засоби синхронізації в OpenMP. Інтерфейс MPI. Необхідні функції. Колективні функції. Комунікаційні операції типу «точка-точка». MPJ Express.
7	Тема 2.1. Огляд java.util.concurrent <u>Основні питання:</u> Синхронайзери. Semaphore. CyclicBarrier. CountdownLatch. Exchanger<V>. Клас Phaser. Locks. Інтерфейсу Lock та Condition. Класи ReentrantLock та ReentrantReadWriteLock.
8	Тема 2.1. Конкурентні колекції в java.util.concurrent <u>Основні питання:</u> CopyOnWrite-колекції. Scalable Maps. Неблокуючі черги.
9	Тема 2.2 Лямбди в мові Java

	<u>Основні питання:</u> Структура лямбда-виразів. Створення лямбда-виразів. Застосування лямбда-виразів. Функціональний інтерфейс Streams. Послідовні та паралельні стріми. Оператори Streams API.
10	Тема 2.2. Застосування CompletableFuture <u>Основні питання:</u> Інтерфейс CompletionStage<T>. Старт обчислень. Отримання результатів. Колбеки. Об'єднання CompletableFuture. Обробка виключень CompletableFuture.
11	Тема 2.3. Реактивне програмування. Технологія RxJava <u>Основні питання:</u> ReactiveX. Патерн «Спостерігач». Клас Observable. Інтерфейс Observer. Гарячі і холодні об'єкти Observable. Типи Observable. Підписка в RxJava. Reactive Streams.
12	Тема 3.1. Низькорівневі мережеві можливості – Sockets <u>Основні питання:</u> Сокети. Класи Socket та ServerSocket. Створення багатопоточних серверів. Рефлексія. Сериалізація. Remote Method Invocation.
13	Тема 3.1. Веб-сервіси SOAP & REST <u>Основні питання:</u> Вебсервіси. SOAP. Web Application Description Language. REST та REST-фреймворки: Jersey, Restlet, RESTEasy. WSDL.
14	Тема 3.1. Мікросервіси <u>Основні питання:</u> Мікросервісна архітектура. Типи мікросервісів. Синхронізація даних між мікросервісами. Використання Spring Boot для розробки мікросервісів.
15	Тема 3.1. Системи роботи з чергами. <u>Основні питання:</u> Типи черг. Java Message Service. Message-Driven Bean. MSMQ. RabbitMQ. Протоколи.
16	Тема 3.2. XML та JSON як альтернативні методи серіалізації <u>Основні питання:</u> JAXB. XmlMapper. ObjectMapper. Jackson та його анотації. XSD-схема. Валідація XML за допомогою XSD-схеми.
17	Тема 3.2. База даних Redis <u>Основні питання:</u> Основні функції. Структури даних Redis. Підключення до Redis. Запити в Redis.
18	Тема 3.3. Розподіленість з Atomix <u>Основні питання:</u> Актори, кластери та розподілений журнал. Розподілені примітиви. Алгоритм консенсусу Raft.

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Практичне заняття №1. Програмування задач із паралелізмом даних. <u>Основні питання:</u> Розподіл даних, координація обчислень
2	Практичне заняття №2. Використання та синхронізація потоків. <u>Основні питання:</u> Потоки в стандарті POSIX, потоки Windows API.
3	Практичне заняття №3. Застосування неблокуючих алгоритмів. <u>Основні питання:</u> Атомарні операції. Алгоритм CAS. Неблокуюча черга. Неблокуючий стек.
4	Практичне заняття №4. Розробка багатопоточних програм. <u>Основні питання:</u> Створення потоків, синхронізація доступу до ресурсів, змагання потоків, Deadlock, Thread-safe код.
5	Практичне заняття №5. Обчислення за допомогою Method Passing Interface. <u>Основні питання:</u> Середовище MPICH.
6	Практичне заняття №6. Застосування асинхронних обчислень. <u>Основні питання:</u> Клас ExecutorService, інтерфейси Future та Callable
7	Практичне заняття №7. Розробка клієнт-серверного додатка. <u>Основні питання:</u> Stream Server, Stream Client, датаграми.

8	Практичне заняття №8. Засоби серіалізації об'єктів в розподілених програмних системах. <i>Основні питання: Java Serialization, Protocol Buffers, JSON.</i>
9	Практичне заняття №9. Робота із системами відправки повідомлень <i>Основні питання: RabbitMQ, відправлення повідомлення, отримання повідомлення.</i>

6. Самостійна робота студента/аспіранта

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Підготовка до практичних занять	54
2	Підготовка до МКР	6
3	Підготовка до заліку	6

6. Контрольні роботи

Метою контрольних робіт є закріплення та перевірка теоретичних знань із кредитного модуля, набуття студентами практичних навичок самостійного вирішення задач.

Одна модульна контрольна робота (МКР) розбивається на дві контрольні роботи тривалістю в одну годину кожна. Контрольні роботи проводяться за допомогою Google Forms під час календарних контролів (атестацій).

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.
- правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
- політика дедлайнів та перескладань: якщо студент не проходив або не з'явиться на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;
- політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Інфографіка в програмному забезпеченні»;
- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.

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

Поточний контроль: тестування, МКР, виконання завдань до практичних занять.

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

Семестровий контроль: залік.

Умови допуску до семестрового контролю: семестровий рейтинг більше 40 балів.
Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
95-100	Відмінно
85-94	Дуже добре
75-84	Добре
65-74	Задовільно
60-64	Достатньо
Менше 60	Незадовільно
Менше 30	Не допущено

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- тестування по кожному лекційному заняттю;
- виконання завдань до практичних занять;
- виконання модульних контрольних робіт (МКР);
- виконання додаткових завдань.

Тестування по лекціям	Практичні заняття	МКР 1	МКР 2	Додаткові бали
18	54	10	10	8

Тестування по матеріалам лекційних занять

Ваговий бал 1. Максимальна кількість балів за тестування – 1 бал * 18 лекцій = 18 балів.

Тестування проводиться за допомогою Google Forms наприкінці поточної лекції. У деяких випадках термін проходження тестування може бути продовжений лектором. Тривалість проходження одного тестування – 5 хвилин. Кількість спроб – одна. У деяких випадках, що пов'язані з технічними проблемами студентів, може надатися повторна спроба на окремі тестування.

Кожне тестування містить 5 запитань різного формату (вибір правильного варіанту з переліку; вибір правильних варіантів з переліку, вірно/невірно; визначити відповідність; чисельна відповідь).

Критерії оцінювання

- запитання типу «вибір правильного варіанту з переліку», «вірно/невірно», «чисельна відповідь» оцінюються однозначно: вірна відповідь – 0,2 бал, невірна відповідь – 0 балів;
- запитання, на які немає однієї конкретної відповіді, типу «вибір правильних варіантів з переліку» або «визначити відповідність» оцінюються у відповідності до кількості елементів у тесті (наприклад, якщо потрібно обрати 4 вірні варіанти відповіді із 7, то студент отримує по 0,05 балів за один правильний варіант відповіді, а за всі 4 правильні відповіді отримує відповідно 0,2 бал) – невірна відповідь – 0 балів, частково вірна відповідь – 0,05-0,15 балів, вірна відповідь 0,2 бал.

Практичні заняття

Ваговий бал – 6. Максимальна кількість балів за всі практичні заняття – 6 бали * 9 занять = 54 балів.

На практичних заняттях студенти розробляють програмний код у відповідності до тематики практичного заняття. Після кожного практичного заняття студенти отримують домашнє завдання, яке необхідно вирішити та надати на перевірку викладачу до початку наступного заняття (зазвичай це 2 тижні, однак іноді цей час може бути змінений викладачем у деяких конкретних випадках).

Критерії оцінювання

- домашнє завдання вирішено вірно та здано протягом 2-х тижнів після практичного заняття – 6 балів;
- домашнє завдання вирішено вірно, але здано протягом більш ніж 2-х тижнів після практичного заняття – 5 балів;
- домашнє завдання вирішено із незначними помилками та здано протягом 2-х тижнів після практичного заняття – 4 бали;
- домашнє завдання вирішено із незначними помилками та здано протягом більш ніж 2-х тижнів після практичного заняття – 3 бали;
- домашнє завдання вирішено із значними помилками – повертається на доопрацювання.

Модульна контрольна робота

Ваговий бал за одну МКР – 10. Максимальний бал за 2 МКР складає 20 балів.

Критерії оцінювання

Кожна частина модульної контрольної роботи складається із 20 завдань. Окреме завдання оцінюється в 0.5 бали.

Календарний контроль базується на поточній рейтинговій оцінці. Умовою позитивної атестації є значення поточного рейтингу студента не менше 50% від максимально можливого на час атестації.

Додаткові (бонусні) бали

Рейтинговою системою оцінювання передбачені додаткові бали за виконання додаткових завдань. Один студент не може отримати більше ніж 8 бонусних балів у семестрі. При отриманні більш ніж 8 балів, вони обмежуються на рівні 8. Бонусні бали можуть бути отримані за виконання «Завдання до лекцій».

Завдання до лекцій. Студенти, за бажанням, можуть виконувати додаткові завдання за матеріалами лекцій. За одне додаткове завдання нараховується 0,5 бали. Максимальна кількість балів, що можна отримати за завдання до лекцій складає 8 балів.

Форма семестрового контролю – залік

Максимальна сума балів складає 100. Необхідною умовою допуску до заліку є зараховані обидві частини МКР та здані усі завдання до практичних занять. Для отримання заліку з кредитного модулю «автоматом» потрібно мати рейтинг не менше 60 балів, а також виконані умови допуску до заліку.

Студенти, які наприкінці семестру мають рейтинг менше 60 балів, а також ті, хто хоче підвищити свою оцінку в системі ECTS, виконують залікову контрольну роботу. При цьому набрані бали студентом анулюються, а оцінка за залікову контрольну роботу є остаточною.

Залікова робота. Залікова робота проводиться на останньому лекційному занятті. Студент проходить тестування за допомогою Google Forms. На тестування пропонується 50 тестових завдань, кожне з яких оцінюється в 2 бали. Для отримання позитивної оцінки необхідно набрати 60 балів і вище. Час тестування зазвичай складає 90 хвилин, але може бути скоригований лектором та (або) викладачем, що приймає залік.

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Класифікація паралельних обчислювальних систем.
2. Типи паралелізму.
3. Оцінка ефективності паралельних алгоритмів.
4. Потoki. Створення потоків.
5. Синхронізація потоків.
6. Принципи синхронізації потоків.
7. Поняття атомарності.
8. Synchronized-методи та блоки.

9. Жорстка синхронізація, семафори, монітори.
10. Поняття дедлоків.
11. Класична задача синхронізації «Обідаючі філософи».
12. Класична задача синхронізації «Читачі та письменники».
13. Класична задача синхронізації «Виробники та споживачі».
14. Класична задача синхронізації про «сплячого перукаря» .
15. Неблокуючі алгоритми.
16. Атомарні операції. Операція атомарної зміни Compare-and-Swap (CAS). ABA-проблема.
17. Алгоритм Скота-Мітчела.
18. Програмна модель OpenMP.
19. Засоби синхронізації в OpenMP.
20. Синхронайзери в java.util.concurrent.
21. Конкурентні колекції в java.util.concurrent.
22. Лямбда-вирази в мові Java.
23. Функціональний інтерфейс. Streams.
24. Stream API.
25. Робота з CompletableFuture.
26. Реактивне програмування. Технологія RxJava.
27. Патерн «Спостерігач».
28. Низькорівневі мережеві можливості – Sockets.
29. Створення багатопоточних серверів.
30. Remote Method Invocation.
31. XML та JSON як альтернативні методи серіалізації.
32. Системи роботи з чергами.JMS. RabbitMQ.
33. Веб-сервіси SOAP та REST.
34. Мікросервіси.
35. Розподіленість з Atomix.

Робочу програму навчальної дисципліни (силабус) «Технології паралельних та розподілених обчислень»:

Складено доцентом кафедри інженерії програмного забезпечення в енергетиці, к.т.н. Варавою І.А.

Ухвалено кафедрою інженерії програмного забезпечення в енергетиці (протокол № 34 від 10.05.2024 р.)

Погоджено Методичною комісією ННІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 31.06.2025 р.)