

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра інтелектуальних програмних систем

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2019 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
КЛАСТЕРНІ РОЗРАХУНКИ

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **121 «Інженерія програмного забезпечення»**
(шифр і назва спеціальності)
освітній рівень **магістр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Програмне забезпечення систем»**
(назва освітньої програми)

вид дисципліни **обов'язкова**

| | |
|--|-------------------|
| Форма навчання | денна |
| Навчальний рік | 2019/2020 |
| Семестр | 1 |
| Кількість кредитів ECTS | 3 |
| Мова викладання, навчання та оцінювання | українська |
| Форма заключного контролю | екзамен |

Викладачі: **к.ф.-м.н., доц. Верес М.М.** (лекції, лабораторні заняття),

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2019

Розробник: Верес Максим Миколайович, к.ф.-м.н., доцент кафедри інтелектуальних програмних систем

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем _____ (Провотар О.І.)
(підпис) (прізвище та ініціали)

Протокол № ____ від «__» _____ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «_____» _____ 20__ року №__

Голова науково-методичної комісії _____ (Омельчук Л.Л.)
(підпис) (прізвище та ініціали)

«_____» _____ 20__ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «_____» _____ 20__ року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – оволодіння навиком самостійного опрацювання науково-технічної літератури в галузі паралельного програмування, методами створення ефективного коду на багатопроцесорних комп'ютерах для наукових розрахунків, застосуванні їх на практиці та самостійної підготовки і проведення наукової презентації на вибрану тему.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності).

1. **Знати:** поняття програмування та принципи розробки програм; класичні алгоритми та різновиди структур даних
2. **Вміти:** проектувати, розробляти та тестувати програми на базовому рівні.
3. **Володіти елементарними навичками:** програмування мовами C, C++.

3. Анотація навчальної дисципліни. Навчальна дисципліна “Кластерні розрахунки” є складовою освітньо-професійної програми підготовки фахівців за другим (магістерським) рівнем вищої освіти галузі знань 12 „Інформаційні технології” зі спеціальності 121 „Інженерія програмного забезпечення”, освітньо-професійної програми „Програмне забезпечення систем”.

Дана дисципліна є нормативною навчальною дисципліною в рамках освітньої програми «Програмне забезпечення систем». Викладається у 1 семестрі 1 курсу магістратури в **обсязі – 90 год. (3 кредити ECTS)** зокрема: лекції – 28 год., самостійна робота – 60 год., консультації – 2 год. У курсі передбачено 2 змістовних частини та 2 контрольні роботи. Завершується дисципліна – **заліком**.

В результаті вивчення навчальної дисципліни студент повинен:

знати відомості про методи побудови паралельних алгоритмів обчислення складних задач на багатопроцесорних комп'ютерах для наукових розрахунків.

вміти проектувати та розробляти паралельні програми на багатопроцесорних комп'ютерах для наукових розрахунків.

Дисципліна «Кластерні розрахунки» є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня «магістр».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні програмування паралельних алгоритмів, відповідно до кваліфікації фахівця з інформаційних технологій. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність проведення теоретичних та прикладних досліджень на відповідному рівні (ЗК03).
- Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК03).
- Вміння планувати і проводити наукові дослідження, готувати результати наукових робіт з інженерії програмного забезпечення до оприлюднення (СК09).

5. Результати навчання за дисципліною:

| Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність) | | Форми (та/або методи і технології) викладання і навчання | Методи оцінювання та пороговий критерій оцінювання (за необхідності) | Відсоток у підсумковій оцінці з дисципліни |
|---|---|--|--|--|
| Код | Результат навчання | | | |
| РН1.1 | Знати поняття програмування на MPI | Лекція, самостійна робота | Тест, 60% правильних відповідей, екзамен | 15% |
| РН1.2 | Знати основні етапи побудови та виконання паралельної програми | Лекція, самостійна робота | Тест, 60% правильних відповідей, залік | 15% |
| РН1.3 | Знати методи побудови паралельних алгоритмів обчислення складних задач на багатопроцесорних комп'ютерах для наукових розрахунків | Лекція, самостійна робота | Тест, 60% правильних відповідей, залік | 20% |
| РН2.1 | Вміти застосовувати на практиці інструментальні програмні засоби проектування та розробки програмного забезпечення. | Самостійна робота | Поточне оцінювання, проектів, залік | 24% |
| РН3.1 | Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти | Самостійна робота | Поточне оцінювання, проектів | 10% |
| РН4.1 | Організувати свою самостійну роботу для досягнення результату | Самостійна робота | Поточне оцінювання, проектів | 8% |
| РН4.2 | Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість | Самостійна робота | Захист проектів | 8% |

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

| Результати навчання дисципліни | РН 1.1 | РН 1.2 | РН 1.3 | РН 2.1 | РН 3.1 | РН 4.1 | РН 4.2 |
|--|--------|--------|--------|--------|--------|--------|--------|
| Програмні результати навчання | | | | | | | |
| <i>(з опису освітньої програми)</i> | | | | | | | |
| ПРН01. Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення. | + | + | + | + | | | + |
| ПРН03. Знати і застосовувати базові концепції і методології моделювання інформаційних процесів. | | + | + | | | | |
| ПРН06. Аналізувати, оцінювати і обирати методи, сучасні | + | + | + | + | | | + |

| | | | | | | | |
|---|--|---|---|---|---|---|---|
| програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії. | | | | | | | |
| ПРН07. Обґрунтовано обирати парадигми і мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення. | | + | | | + | + | + |
| ПРН08. Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення. | | | + | + | + | + | + |

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольна робота 1: РН 1.1, РН 1.2 — 10 балів/6 балів.
2. Контрольна робота 2: РН 1.2, РН1.3 — 10 балів/6 балів.
3. Самостійна робота (проект 1): РН 1.2, РН1.3, РН 2.1, РН3.1 – 20 балів/10 бали.
4. Самостійна робота (проект 2): РН1.2, РН1.3, РН 2.1, РН3.1 – 20 балів/10 бали.
5. Самостійна робота (проект 3): РН1.2, РН1.3, РН 2.1, РН3.1 – 20 балів/10 бали.
6. Самостійна робота (проект 4): РН1.2, РН1.3, РН 2.1, РН3.1 – 20 балів/10 бали.

Відповідно до пункту 7.1.5 Положення про організацію освітнього процесу у Київському національному університеті імені Тараса Шевченка (<http://knu.ua/pdfs/official/Organization-of-the-educational-process.pdf>) якщо формою підсумкового контролю є залік, то він виставляється як сума усіх балів по усіх компонентах передбачених програмою дисципліни, і ця сума у максимальному випадку рівна 100 балів, у мінімальному - 60. Також недопуск до заліку - лише у випадку, якщо студент набрав менше 60% від максимально-можливої кількості балів.

Щодо недопуску до іспиту, то студент - недопущений до іспиту лише якщо він протягом семестру не набрав хоча б 50% від максимально-можливої кількості балів, тобто якщо у семестрі він може набрати максимум 60 балів, то мінімум - це 30 балів, а максимальний мінімум - це 36 балів (60% від 60 балів).

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота : до 7 тижня семестру.
2. Контрольна робота : до 14 тижня семестру.
3. Самостійна робота (проект 1): до 3 тижня семестру.
4. Самостійна робота (проект 2): до 6 тижня семестру.
5. Самостійна робота (проект 3): до 9 тижня семестру.
6. Самостійна робота (проект 4): до 12 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного виконання проекту, викладач має право не зарахувати проект, або знизити бали.

Студент має право здавати проекти після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

| | |
|----------------------------------|--------|
| Відмінно / Excellent | 90-100 |
| Добре / Good | 75-89 |
| Задовільно / Satisfactory | 60-74 |
| Незадовільно / Fail | 0-59 |
| Зараховано / Passed | 60-100 |
| Не зараховано / Fail | 0-59 |

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

| № лекції | Назва лекції | Кількість годин | | |
|---|--|-----------------|-------------|------------|
| | | Лекції | Лабораторні | Самостійна |
| Частина 1. Мета та задачі паралельної обробки даних. Класифікація. Моделі обчислень та методи аналізу ефективності. Паралельні алгоритми розв'язку задач | | | | |
| 1 | Тема 1 Мета та задачі паралельної обробки даних. | 2 | | 4 |
| 2 | Тема 2. Зв'язок між елементами паралельних систем розрахунку. Топологія. | 2 | | 4 |
| 3 | Тема 3. Класифікація багатопроесорних систем | 2 | | 4 |
| 4 | Тема 4. Векторно-конвейерні суперкомп'ютери. Кластерні системи. | 2 | | 4 |
| 5 | Тема 5. Моделі обчислень та методи аналізу ефективності. | 2 | | 4 |
| 6 | Тема 6. Аналіз комунікаційної складності паралельних алгоритмів. | 2 | | 4 |
| 7 | Тема 7. Принципи розробки паралельних методів. Паралельні матричні обчислення. | 2 | | 4 |
| 8 | Тема 8. Паралельне сортування. Паралельні алгоритми розв'язку систем лінійних рівнянь | 2 | | 4 |
| | Контрольна робота 1 | | | |
| Всього по частині 1 | | 18 | | 32 |
| Частина 2. Інструментальні засоби паралельного програмування. Паралельні алгоритми розв'язку задач | | | | |
| 9 | Тема 9. Технологія розробки для багатопроесорних систем с розподіленою пам'яттю (стандарт передачі повідомлень MPI) | 2 | | 8 |
| 10 | Тема 10. Технологія паралельного програмування PVM. Технологія паралельного програмування DVM | 2 | | 4 |
| 11 | Тема 11. Технологія розробки для багатопроесорних систем с розділеною пам'яттю (стандарт OpenMP). | 2 | | 4 |
| 12 | Тема 12. Паралельні алгоритми на графах | 2 | | 4 |
| 13 | Тема 13. Паралельний розв'язок диференційних рівнянь | 2 | | 4 |
| 14 | Тема 14. Концепція Grid та метакомп'ютинг. | 2 | | 4 |
| | Контрольна робота 2 | | | |
| Всього по частині 2 | | 10 | | 30 |
| | Консультація | 2 | | |
| | Екзамен | | | |
| | ВСЬОГО | 28 | | 60 |

Загальний обсяг 90 год., в тому числі:

Лекцій – **28** год.

Консультації – **2** год.

Самостійна робота - **60** год.

Теми, винесені на самостійне вивчення:

1. Продуктивність паралельних комп'ютерів.
2. Порівняння продуктивності архітектури ПК при різних задачах.
3. Критерій оцінки топології мережі .
4. Алгоритми маршрутизації і методи передачі даних.
5. Типові операції взаємодії.
6. Представлення топології мережі.
7. Етапи розробки: розділення обчислень, виділення інформаційних залежностей, масштабування і розподілення обчислень між процесами

Виконання проектів 1-4(аналіз прискорення та ефективності).

1. Паралельні алгоритми сортування.
2. Паралельні алгоритми розв'язку систем лінійних рівнянь
3. Паралельні алгоритми на графах
4. Паралельні алгоритми розв'язку диференційних рівнянь

Деталізовані умови лабораторних робіт розміщено за посиланнями:

<https://drive.google.com/drive/folders/0B1dkotQCgQPvZTJ0d2lvcEpPbk0>

9. Рекомендовані джерела:

Основні

1. Foundations of Multithreaded, Parallel, and Distributed Programming. - Gregory R. Andrews Addison-Wesley, 2000. ISBN: 0201357526, 9780201357523.
2. Distributed Systems. Maarten van Steen, Andrew S. Tanenbaum, 2017. ISBN:9781543057386, 1543057381
3. Distributed Systems. Principles and Paradigms. Andrew S. Tanenbaum, Maarten van Stehen, 2016. ISBN:9781530281756, 153028175X
4. Using MPI, Parallel Programming with the Message-Passing Interface. William Gropp, Ewing Lusk, Anthony Skjellum, 2014. ISBN:9780262527392, 0262527391
5. Parallel Programming in MPI and OpenMP. Victor Eijkhout, 2017. ISBN:9781387400287, 1387400282

Додаткові:

1. <https://www.mpich.org/>
2. <http://www.mpi-forum.org>
3. <http://openmp.org>
4. <https://learn.microsoft.com/en-us/message-passing-interface/microsoft-mpi>
5. <https://www.ee.ryerson.ca/~courses/ee8218/>
6. https://turing.une.edu.au/~cosc330/lectures/root_directory.html
7. <https://blogs.msdn.microsoft.com/ru-hpc/2009/12/28/ms-mpi-visual-studio-windows-hpc-server/>
8. <https://azure.microsoft.com/en-us/documentation/articles/cloud-services-setup-hybrid-hpcpack-cluster/>