

**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
РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ
для студентів**

галузь знань **12 «Інформаційні технології»**
спеціальність **121 «Інженерія програмного забезпечення»**
освітній рівень **бакалавр**
освітня програма **«Програмна інженерія»**
вибірковий блок **«Програмна інженерія»**
вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	6
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

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

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.

на 20__/20__ н. р. _____ (_____) «__» 20__ р.

КИЇВ – 2019

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

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

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

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – оволодіння концепціями сучасного програмування в рамках парадигм паралельного та розподіленого програмування, в тому числі з роздільними і розподільними змінними, ефективне розпаралелення послідовних алгоритмів програм. Основу вивчення складають підходи до програмування в багатопотокових системах, розподілених системах, системах синхронних паралельних розрахунків. Розглядаються проблеми сумісної роботи процесів паралельної програми та їх синхронізації.

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

1. **Знати:** основні поняття об'єктно-орієнтованого програмування, принципи розробки програм; базові класичні алгоритми та різновиди структур даних.
2. **Вміти:** проектувати, розробляти та тестувати програми на базовому рівні.
3. **Володіти навичками ООП:** програмування мовами C, C++, Java.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Розподілені обчислення» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення», освітньо-професійної програми «Програмна інженерія».

Дана дисципліна є освітнім компонентом вибіркового блоку «Програмна інженерія» освітньо-професійної програми «Програмна інженерія». Викладається у **5 семестрі 3 курсу в обсязі – 120 год. (4 кредитів ECTS)** зокрема: лекції – 28 год., лабораторні – 28 год., самостійна робота – 62 год., консультації – 2 год. У курсі передбачено **2 змістовні частини та 2 контрольні роботи. Завершується дисципліна – іспитом.**

В результаті вивчення навчальної дисципліни студент повинен:

знати відомості про методи побудови паралельних алгоритмів обчислення в багатопотокових системах, розподілених системах, системах синхронних паралельних розрахунків.

вміти проектувати та розробляти паралельні програми на багатоядерних та багатопроцесорних комп'ютерах.

Дисципліна «Розподілені обчислення» є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня «бакалавр» програмістського спрямування програми «Програмна інженерія».

4. Завдання (навчальні цілі):

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

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність розробляти архітектури, модулі та компоненти програмних систем (СК03).

- Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. (СК07).
- Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (СК13).
- Здатність до алгоритмічного та логічного мислення (СК14).
- Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень. (СК16.2).
- Здатність застосовувати знання архітектури та схемо-технічних основ сучасних комп'ютерів (СК17.2) .

ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	+	+		+	+	+	+
ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.	+	+	+				
ПРН06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.	+	+	+			+	+
ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення	+	+	+	+			+
ПРН10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.				+	+	+	+
ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+	+	+			+	+
ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.	+	+	+				
ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.	+	+		+		+	+
ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних				+			
ПРН25.2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для розв'язання завдань інженерії програмного забезпечення.			+			+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольна робота 1: РН 1.1, РН 1.2, РН4.2 — **10 балів/6 балів.**
2. Контрольна робота 2: РН 1.2, РН1.3, РН4.2 — **10 балів/6 балів.**
3. Лабораторна робота 1: РН 2.1, РН3.1, РН4.1, РН4.2 — **8 балів/5 балів.**
4. Лабораторна робота 2: РН 2.1, РН3.1, РН4.1, РН4.2 — **6 балів/4 бали.**
5. Лабораторна робота 3: РН 2.1, РН3.1, РН4.1, РН4.2 — **6 балів/4 бали.**
6. Лабораторна робота 4: РН 2.1, РН3.1, РН4.1, РН4.2 — **6 балів/4 бали.**
7. Лабораторна робота 5: РН 2.1, РН3.1, РН4.1, РН4.2 — **6 балів/4 бали.**
8. Лабораторна робота 6: РН 2.1, РН3.1, РН4.1, РН4.2 — **8 балів/5 балів.**

Підсумкове оцінювання (у формі іспиту) вказується:

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН4.1, РН4.2;
- форма проведення і види завдань: письмова робота.

Види завдань: 5 письмових завдань.

Критерії оцінювання на іспиті

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1-2	Теоретичне питання з розподіленого програмування	По 20%	40%
Завдання 3-5	Задача з розподіленого програмування	По 20%	60%
			100%

Типове завдання контрольної роботи 1.

Розробити багатопоточний консольний додаток з використанням об'єктів синхронізації.

Порт. Кораблі заходять в порт для розвантаження / завантаження контейнерів, Число контейнерів, що знаходяться в поточний момент в порту і на кораблі, має бути невід'ємним і перевищувати задану вантажопідйомність судна і місткість порту. У порту працює кілька причалів. У одного причалу може стояти один корабель. Корабель може завантажуватися біля причалу, розвантажуватися або виконувати обидві дії.

Типове завдання контрольної роботи 2.

Реалізувати за допомогою технології клієнт-сервер програму На клієнті вибираємо функцію $f(x) = \{a \cdot \sin(x), a + \cos(x), \tan(x) - a, ax + b\}$, відправляємо на сервер запит на обчислення цієї функції та діапазон значень з кроком обчислень. Сервер повертає масив значень, які клієнт відображає у вигляді таблиці $| x | f(x) |$.
а) використовуючи Socket (сервер багатопоточний).
б) використовуючи RMI.

Студенти не допускаються до іспиту, якщо під час семестру вони набрали менше ніж 36 балів та/або не виконали всі передбачені планом роботи.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота : до 7 тижня семестру.
2. Контрольна робота : до 14 тижня семестру.
3. Лабораторна робота 1: до 3 тижня семестру.
4. Лабораторна робота 2: до 5 тижня семестру.
5. Лабораторна робота 3: до 7 тижня семестру.
6. Лабораторна робота 4: до 10 тижня семестру.
7. Лабораторна робота 5: до 12 тижня семестру.
8. Лабораторна робота 6: до 14 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 лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи 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

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторні	Самостійна
Частина 1. Модель, спеціалізації паралельних та розподілених розрахунків. Багатопоточне та паралельне програмування з використанням спільної пам'яті.				
1.	Тема 1. П'ять головних парадигм паралельного програмування	2	2	4
2.	Тема 2. Програмування з спільною та розподіленою пам'яттю	2	2	4
3.	Тема 3. Процеси та синхронізація. Моделі взаємодії процесів.	2	2	4
4.	Тема 4. Критичні секції. Семафори, монітори	2	2	4
5.	Тема 5. Бар'єри, блокування читання запису, паралельні колекції	2	2	4
6.	Тема 6. Мета та задачі паралельної обробки даних. Моделі обчислень та методи аналізу ефективності. Принципи розробки паралельних алгоритмів.	2	2	6
7.	Тема 7. Паралельні алгоритми розв'язку задач. Матричні операції.	2	2	4
	Контрольна робота 1			2
Контроль за підсумками лабораторних робіт 1,2 та 3				
Всього по частині 1		14	14	32
Частина 2. Задачі паралельної обробки даних з використанням розподіленої пам'яті. Паралельні та розподілене програмування.				
8.	Тема 8. Технологія розробки для багатоядерних та багато процесорних систем (стандарт передачі повідомлень MPI та OpenMP)	2	2	4
9.	Тема 9. Формати збереження та передачі даних по мережі(бінарна серіалізація, XML,JSON)	2	2	4
10.	Тема 10. Модель клієнт-сервер. Передача даних за допомогою сокетів (Socket).	2	2	4
11.	Тема 11. Віддалений виклик процедур, рандеву(RMI).	2	2	4
12.	Тема 12. Асинхронна передача даних за допомогою MQ(JMS).	2	4	4
13.	Тема 13. Web програмування(JSP, Servlet)	2	2	4
14.	Тема 14. Web сервіси(SOAP, REST)	2		4
	Контрольна робота 2			2
Контроль за підсумками лабораторних робіт 4, 5 та 6				
Всього по частині 2		14	14	30
Консультація			2	
Іспит				
ВСЬОГО		28	28	64

Загальний обсяг 120 год., в тому числі:

Лекцій – 28 год.

Лабораторні заняття - 28 год.

Консультацій – 2 год.

Самостійна робота - 62 год.

Теми, винесені на самостійне вивчення:

1. Модель OSI.
2. Архітектури мережі.
3. Алгоритми маршрутизації.
4. Типові топології побудови розподілених систем.
5. Технології Corba
6. .NET Remoting, WCF
7. RPC
8. MOM
9. OpenMP
10. IntelTBB
11. TPL
12. MS#

Виконання лабораторних робіт 1-6.

1. Багатопоточне програмування з використання об'єктів синхронізації
2. Багатопоточне програмування в графічному інтерфейсі
3. Паралельні алгоритми обчислення матриць
4. Однорівнева система. Створити додаток з графічним інтерфейсом в якому реалізувати операції CRUD
5. Дворівнева система клієнт-сервер. Клієнт - графічна форма, на серверній частині DAO клас взаємодії з DBMS.
6. Багаторівнева система з використанням MQ та Web інтерфейсом.

Деталізовані умови лабораторних робіт розміщено за посиланням:

<https://drive.google.com/drive/folders/0B1dkotQCgQPvWIk1NzZicEhFR00>

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 Steen, 2016. ISBN:9781530281756, 153028175X
4. Java: The Complete Reference, Eleventh Edition By Herbert Schildt, 2018. ISBN:9781260440249, 1260440249
5. Java Concurrency in Practice. - Addison-Wesley Professional, 2006. ISBN:9780321349606, 0321349601
6. Programming in Go. Mark Summerfield, 2012. ISBN:9780132764094, 0132764091.
7. Distributed Computing with Go. V.N. Nikhil Anurag, 2018. ISBN:9781787127708, 1787127702
8. Modern Java in Action: Lambdas, streams, functional and reactive programming. - Manning Publications, 2018.

Додаткові:

1. <http://www.mpi-forum.org>
2. <http://openmp.org>
3. <https://docs.oracle.com/javase/tutorial/>
4. <https://go.dev/doc>
5. https://golang.org/doc/effective_go.html#channels
6. https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/basic_app_embedded_tomcat/basic_app-tomcat-embedded.html
7. <http://www.intel.com/software/>
8. <http://www.mcsharp.net/>