

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ  
КАФЕДРА МАТЕМАТИЧНОЇ ІНФОРМАТИКА**

**«ЗАТВЕРДЖУЮ»**  
Заступник декана  
з навчальної роботи  
Олена КАШПУР  
« 12 » лютого 2021 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
СИСТЕМИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ**

для студентів

галузь знань **12 «Інформаційні технології»**  
спеціальність **122 «Комп'ютерні науки»**  
освітній рівень **бакалавр**  
освітня програма **«Інформатика»**  
вид дисципліни **вибіркова**  
вибірковий блок **«Інтелектуальні інформаційні технології»**

Форма навчання	<b>денна</b>
Навчальний рік	<b>2021/2022</b>
Семестр	<b>8</b>
Кількість кредитів ECTS	<b>4</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>іспит</b>

**Викладачі:** к.ф.-м.н., доцент Деревянченко О.В. (лекції, лабораторні заняття).

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


**КИЇВ – 2021**

© Деревянченко О.В., 2021 рік

Розробник: Дервянченко Олександр Валерійович, кандидат фіз.-мат. наук, доцент кафедри математичної інформатики


ЗАТВЕРДЖЕНО

Завідувач кафедри математичної інформатики

 Василь ТЕРЕЩЕНКО

Протокол № 6 від «11» 02 2021 р.

Схвалено Гарантом освітньо-професійної програми «Інформатика»

«11» лютого 2021р.  Людмила ОМЕЛЬЧУК

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

Протокол від «11» лютого 2021 року № 7

Голова науково-методичної комісії  Людмила ОМЕЛЬЧУК

«11» лютого 2021 року

**1. Мета дисципліни «СИСТЕМИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ»** є важливою дисципліною в сенсі набуття теоретичних та практичних знань в одній з найактуальніших на сьогодні галузей інформаційних технологій, що стосується програмування складних за обсягом обчислень задач з застосуванням розподілених середовищ (хмари, кластери).

## **2. Попередні вимоги до опанування або вибору навчальної дисципліни:**

1. Успішне опанування курсів «Основи програмування», «Програмування» та «Основи об'єктно-орієнтованого програмування», «Інструментальні середовища та технології програмування», «Архітектура обчислювальних систем та комп'ютерні мережі».
2. Знати теоретичні основи теорії алгоритмів та математичної логіки.
3. Володіння технологіями програмування, основні відомості з програмування на мові JAVA.

## **3. Анотація навчальної дисципліни:**

Навчальна дисципліна «СИСТЕМИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ» є складовою освітньо-професійної програми підготовки фахівців за освітньо-кваліфікаційним рівнем «бакалавр» галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні», освітньо-професійної програми «Інформатика».

Метою і завданням навчальної дисципліни є ознайомлення з одним із основних науково-технологічних напрямків у галузі комп'ютерних технологій - хмарних технологій для опанування знань, що готують студентів для професійної діяльності з розгортання та налаштування хмарних сервісів та оволодіння хмарними технологіями в контексті паралельної та розподіленої обробки великих об'ємів даних.

Дана дисципліна є вибірковою навчальною дисципліною за **програмою «Інформатика»**.

Викладається у 8 семестрі в **обсязі – 120 годин.**

**(4 кредитів ECTS)** зокрема: *лекції – 38 год., консультації – 2 год., самостійна робота – 80 год.* Завершується дисципліна **іспитом у 8 семестрі.**

## **4. Завдання (навчальні цілі):**

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні та обробці великих об'ємом даних, відповідно освітньої кваліфікації «Бакалавр інженерії програмного забезпечення». Зокрема, розвивати:

- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктноорієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж;
- здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації;
- здатність до алгоритмічного мислення(СК19.1);
- здатність застосовувати знання у практичних ситуаціях.

## 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 основні поняття та області паралельних обчислювальних систем. Інфраструктуру та можливості хмарних сервісів.	<i>Лекція</i>	<i>Контрольна робота, 60% правильних відповідей, доповідь</i>	25%
РН1.2	Знати принципи систем паралельного програмування та розгортання у хмарному середовищі. Питання безпеки, масштабування, та резервного копіювання.	<i>Лекція</i>	<i>Контрольна робота, 60% правильних відповідей, доповідь</i>	25%
РН1.3	Знати засоби побудови та розгортання ПАРКС-Java системи у хмарах та основні принципи хмарних обчислень із застосуванням технології ПАРКС.	<i>Лекція, самостійна робота</i>	<i>Захист лабораторної роботи</i>	25%
РН2.1	Вміти оцінювати і вибирати необхідні методи та технології для вибору системи (технології) паралельного програмування та вирішення поставленої задачі.	<i>Самостійна робота</i>	<i>Захист лабораторної роботи</i>	15%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти.	<i>Самостійна робота</i>	<i>Захист лабораторної роботи</i>	5%
РН4.1	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість.	<i>Самостійна робота</i>	<i>Захист лабораторної роботи</i>	5%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Програмні результати навчання	Результати навчання дисципліни					
	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 3.1	РН 4.1
<i>(з опису освітньої програми)</i>						
<b>ПРН19.1.</b> Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+	+	+	+	+	+

## **7. Схема формування оцінки.**

### **7.1. Форми оцінювання студенті**

#### **- семестрове оцінювання:**

1. *Контрольна робота 1:* РН1.1 – 5 балів/3 бали;
2. *Контрольна робота 2:* РН1.2 – 5 балів/3 балів;
3. *Виконання завдань самостійної роботи та Доповідь:* РН1.1 , РН1.2, РН2.1, РН3.1, РН4.1 – 20 балів/12 балів;
4. *Виконання лабораторної роботи:* РН1.3, РН2.1, РН3.1, РН4.1 – 30 балів/18 балів.  
*Максимальна кількість балів, які можуть бути отримані студентом: 60 балів.*

#### **- підсумкове оцінювання (у формі іспиту):**

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

*Види завдань:* 4 теоретичних запитань максимально по 10 балів за кожне.

Критерії оцінювання відповіді студента на теоретичне питання:

- повнота розкриття питання 1-8 бали;
- логіка викладення 1 бал;
- аналітичні міркування 1 бали.

***Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 20 балів.***

#### **Запитання для підготовки до іспиту**

1. Паралельні обчислювальні системи, властивості, приклади.
2. Класифікація систем паралельних обчислень.
3. Показники продуктивності паралельних обчислень: прискорення, ефективність та вартість обчислень.
4. Суперкомп'ютери та кластери.
5. Обчислення на базі GRID.
6. Чим відрізняються парадигми паралельного програмування?
7. Які ви знаєте параметри архітектури мультипроцесорних систем?
8. Які існують шляхи для масштабування та переносимості паралельного програмного забезпечення ?
9. Які види паралелізму використовуються в архітектурі паралельних ЕОМ?
10. Яка концепція керуючого простору та алгоритмічного модулю?
11. Що таке ПАРКС-технологія програмування?
12. Засоби робота з системою ПАРКС-JAVA.
13. Засоби робота з системою ПАРКС-Python.
14. Засоби робота з системою ПАРКС-C#.
15. Розподіл ресурсів та алгоритми диспетчеризації(балансування) в СПОІ(системах паралельної обробки інформації).
16. Існуючі ПАРКС системи, порівняння.
17. Гетерогенний ПАРКС, особливості та можливості.
18. Застосування технології паралельного програмування зі спільною пам'яттю.
19. Застосування технології паралельного програмування з розподіленою пам'яттю.
20. Застосування кластерів для обчислення паралельних програм.
21. Застосування GPU для паралельного програмування.

22. Порівняння різних підходів паралельного програмування.
23. Які існують технології паралельного програмування?
24. Гетерогенний ПАРКС, особливості налаштування.
25. Застосування IBM Natural Language Understanding
26. Застосування Azure Synapse Analytics
27. Застосування AWS S3
28. Застосування AWS CodeGuru
29. Застосування Google Cloud NL API
30. Застосування AWS Textract
31. Застосування DataBricks
32. Застосування Storj Labs
33. Застосування Google Cloud SQL
34. Застосування Tableau
35. Застосування Amazon DynamoDB
36. Застосування AWS Transcribe
37. Застосування Serverless API with AWS.
38. Застосування Docker у Cloud
39. Застосування AWS Lex
40. Застосування Google Firebase (Cloud Storage - ML)
41. Застосування AWS Comprehend
42. Застосування Ареер
43. Застосування AWS Translate
44. Застосування AI Platform
45. Застосування BigQuery
46. Developing Applications with Google Cloud Platform
47. Застосування Google Compute Engine & Virtual Machine
48. Застосування Google App Engine.
49. Застосування Cloud Storage.
50. Застосування Google Container Engine.
51. Застосування Google Kubernetes Engine.

## 7.2. Організація оцінювання:

Обов'язковим для заліку є виконання усіх контрольних робіт та лабораторної роботи до вказаної викладачем дати, перед початком екзаменаційної сесії, згідно навчального плану. Переписування чи перескладання тем не практикується. Дозволяється здача окремих завдань тем у проміжках між написанням контрольних робіт (наприклад, перша тема здається до здачі наступної контрольної роботи у будь-який зручний для викладача та студента час).

### **Терміни проведення форм оцінювання:**

1. Виконання завдань, винесених на самостійну роботу: протягом семестру;
2. Контрольна робота 1: до 3 тижня семестру;
3. Контрольна робота 2: до 7 тижня семестру;
4. Підготовка доповіді (за темами Доповідей та лекційних занять та самостійної роботи): до 9 тижня;
5. Виконання лабораторної роботи: до 10 тижня семестру.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

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

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про організацію освітнього процесу у Київському національному університеті імені Тараса Шевченка”.

### **При визначені оцінки визначальною є робота в семестрі.**

Протягом семестру кожен студент повинен виконати лабораторні роботи та звіт за ними. Всі роботи мають однакову складність і відрізняються лише предметною областю, яку студент обирає на власний смак і узгоджує з викладачем. Сумарна оцінка за практичний модуль(лабораторна робота) та доповідь складає відповідно: 30+20=50 балів.

### **Теми для доповідей**

**Доповідь (презентація на 15-20 слайдів, доповідь на 25-30 хвилин) на одну з тем:**

1. Google Cloud Storage.
2. Google Cloud Translation API.
3. Google Cloud Vision API.
4. Google Cloud Speech-to-Text API.
5. Google Cloud Text-to-Speech API.
6. Google Video Intelligence API.
7. Google Colab for ML.
8. Google Cloud Firestore.
9. Google Firebase.
10. Google Vertex AI.
11. Google Cloud CDN.
12. Google Big Query
13. Google Cloud SQL
14. Azure Custom Vision
15. Automated ML in Azure.
16. Azure Databricks.
17. Azure Synapse Analytics
18. AWS Transcribe.

19. AWS Step functions.
20. AWS Translate.
21. AWS Textract.
22. AWS Rekognition.
23. AWS IoT Core.
24. AWS S3.
25. AWS CodeGuru.
26. AWS Lex.
27. AWS Serverless API.
28. AWS Comprehend.
29. Amazon DynamoDB.
30. Dialogflow API.
31. MongoDB Atlas.
32. Docker.
33. Google Kubernetes Engine.

**Форма звіту:** презентація на 15-20 слайдів, доповідь та практичне застосування. Інформацію записуємо на Google Диск Вашої групи в персональну папку.

### Практичний модуль

*Вимоги до лабораторних робіт:*

#### Завдання

Обирається з класу задач, що потребує паралельної та розподіленої обробки. Студент обирає предметну область на власний смак і узгоджує з викладачем.

#### Дані

1) Всі дані повинні вводитись з попередньо згенерованого файлу, необхідно мати декілька наборів даних(файлів) для тестування.

#### Реалізація та Тестування

Для реалізації поставленої задачі обирається Завдання, яке спочатку треба реалізувати «попередньо» за допомогою однієї з мов програмування Python або Java,, а потім за допомогою відповідної системи ПАРКС на віртуальних машинах (воркерів) у Cloud.

#### Звіт

Роздруковувати не потрібно інформацію записуємо на Google Диск Вашої групи в персональну папку.

В звіті мають бути наступні розділи:

*Опис завдання* – алгоритм, що реалізує поставлену задачу.

*Посилання на реалізацію(код)* – репозиторій <https://github.com/> або інший з відкритим доступом.

*Вимоги до користувача* – опис засобів запуску та результати тестування програми: заносимо у відповідну таблицю.

*У Висновках* потрібно порівняти різні підходи до реалізації Вашої Задачі в Лабораторній роботі.

Робота здається в робочому стані, всі складові мають нормально працювати; окрім цього, викладач буде пропонувати провести тестування та перевіряти розуміння аспектів реалізації(щодо технології, яка використовується ) та алгоритму поставленого Завдання.

### 7.3. Шкала відповідності

<b>Відмінно / Excellent</b>	90-100
<b>Добре / Good</b>	75-89
<b>Задовільно / Satisfactory</b>	60-74
<b>Незадовільно / Fail</b>	0-59

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабор. заняття	Самост. робота
<b>Частина I. Теорія розподілених систем та хмарних обчислень</b>				
1	<b>ТЕМА 1.</b> Вступна частина. Поява та розвиток паралельних обчислювальних систем. Архітектури систем паралельних обчислень. Переваги та недоліки.	2		2
2	<b>ТЕМА 2.</b> Архітектури хмарних систем. Основні переваги та недоліки.	2		2
3	<b>ТЕМА 3.</b> Основні типи віртуалізації.	2		2
4	<b>ТЕМА 4.</b> Кластери та їх застосування. Переваги та недоліки.	2		2
5	<b>ТЕМА 5.</b> Обчислення на базі GRID. Переваги та недоліки.	2		2
Контрольна робота 1		1		
<b>Частина II. Хмарні технології</b>				
6	<b>ТЕМА 6.</b> Огляд сучасних хмарних платформ: Google Cloud Platform, Amazon Web Service, Microsoft Azure. Основні компоненти та засоби.	2		4
7-9	<b>ТЕМА 7.</b> Огляд хмарних застосунків для систем машинного навчання та штучного інтелекту.	4		16
10	<b>ТЕМА 8.</b> Розгортання та масштабування веб-застосунків у хмарних платформах.	2		4
11	<b>ТЕМА 9.</b> Хмарні сховища даних та їх застосування при проектуванні розподілених систем паралельної обробки.	2		4
12	<b>ТЕМА 10.</b> Безпека даних при використанні хмарних сервісів. Сучасні підходи.	2		6
Контрольна робота 2		1		
<b>Частина III. ПАРКС-технологія програмування для паралельних обчислень</b>				
13	<b>ТЕМА 11.</b> Застосування ПАРКС-технології програмування для хмарних обчислень.	2		4
14-15	<b>ТЕМА 12.</b> Розподіл ресурсів та алгоритми диспетчеризації. Робота з системою ПАРКС-JAVA у Cloud.	2		12
16	<b>ТЕМА 13.</b> Огляд засобів для паралельних обчислень у Cloud на GPU.	2		8
17-18	<b>ТЕМА 14.</b> Гетерогенний ПАРКС, можливості та особливості налаштування.	4		8
19	<b>ТЕМА 15.</b> Порівняння різних підходів з використанням Cloud для розподілених паралельних обчислень.	2		4
Лабораторна робота		2		

ВСЬОГО	38	80
--------	----	----

*Загальний обсяг 120 годин, в тому числі:*

Лекцій – 38 год.

Консультації – 2 год.

Самостійна робота – 80 год.

## 9. Рекомендовані джерела:

*Основні:*

1. Деревянченко О.В. Налаштування системи ПАРКС для хмарних обчислень // Навчальний посібник, Київ, 2017-2020 р., 60 с. [Електронний ресурс] – Режим доступу до ресурсу: <http://parcs.unicyb.kiev.ua/mr/>
2. Федоров А. Г. Windows Azure: облачная платформа Microsoft / А. Г. Федоров, Д. Н. Мартынов. [Электронный ресурс]. –Режим доступа: <http://kak.znate.ru/docs/index-61012.html>. – Дата доступа: 03.10.2016. Системы параллельной обработки. // Под ред. Д. Ивенса. М. ,Мир,1985,416 с.
3. Технології паралельного програмування - [Електронний ресурс] – Режим доступу до ресурсу: <http://parallel.ru/tech/>
4. Анисимов А.В., Деревянченко А.В. Система ПАРУС-JAVA для параллельных вычислений на компьютерных сетях // Кибернетика и системный анализ, 2005, №1, С.25-36.
5. Деревянченко О.В. Моделювання паралельних програм за допомогою системи ПАРКС-JAVA // Наукові записки НаУКМА, Комп'ютерні науки. 2005, Т.36., С.32-38.
6. В. Eckel, Thinking in Java // Prentice Hall, 1999, 859 p.
7. Белицкий Р.И., Логинов В.П. Выбор алгоритма диспетчеризации для мультипроцессорной системы с общей шиной // Управляющие системы и машины,1991, №1, С. 9–13.
8. Деревянченко А.В. Алгоритм оптимизации распределения заданий для решения параллельных задач в неоднородной мультипроцессорной системе // Материалы VI международного научно-практического семинара. 2006, Т.1, С.175-180.
9. Google Cloud Platform - [Електронний ресурс] – Режим доступу до ресурсу: <https://console.cloud.google.com/>
10. Деревянченко О.В. ПАРКС-JAVA система для паралельних обчислень на комп'ютерних мережах // Навчальний посібник, Київ, 2011 р., 60 с. [Електронний ресурс] – Режим доступу до ресурсу: <http://www.csc.knu.ua/en/library/books/derevianchenko-35.pdf>

*Додаткові:*

1. S. Akhter, J. Roberts. Multi-Core Programming. — Intel Press, 344p.
2. Richard Gerber, Aart J.C. Bik, Kevin B. Smith, and Xinmin Tian The Software Optimization Cookbook, Second Edition — Intel Press, 404p.
3. Czarnul P. Parallel Programming for Modern High Performance Computing Systems// CRC Press, 2018. – 304p.
4. Kurgalin S., Borzunov S. A Practical Approach to High-Performance Computing// Springer, 2019. – 206 p.
5. С.А. Немнюгин Методические материалы по курсу "Средства программирования для многопроцессорных вычислительных систем"- [Електронний ресурс] – Режим доступу до ресурсу:

[http://www.phys.spb.ru/content/File/Library/studentlectures/Nemnugin/Metod\\_Nemnugin\\_Intel.pdf](http://www.phys.spb.ru/content/File/Library/studentlectures/Nemnugin/Metod_Nemnugin_Intel.pdf)

6. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М., Мир, 1979.