

Чернівецький національний університет імені Юрія Федьковича  
Факультет математики та інформатики  
Кафедра математичного моделювання



**“ЗАТВЕРДЖУЮ”**

**Декан факультету  
математики та інформатики**

**/Мартинюк О.В./**

**06 2024 року**

**РОБОЧА ПРОГРАМА  
навчальної дисципліни**

**Технології розробки розподілених баз даних  
обов'язкова**

<b>Освітньо-професійна програма</b>	Системний аналіз
<b>Спеціальність</b>	124 – Системний аналіз
<b>Галузь знань</b>	12 – Інформаційні технології
<b>Рівень вищої освіти</b>	другий (магістерський)
	Факультет математики та інформатики
<b>Мова навчання</b>	українська

**Чернівці 2024**

Робоча програма навчальної дисципліни "Технології розробки розподілених баз даних" складена відповідно до освітньо-професійної програми "Системний аналіз", затвердженої 29 травня 2023 р., протокол №5.

Розробник: Олександр Матвій,  
доцент кафедри математичного моделювання,  
кандидат фіз.-мат. наук, доцент

Погоджено з гарантом ОП і затверджено на засіданні кафедри математичного моделювання.

Протокол № 18 від 25 червня 2024 року.

Завідувач кафедри

  
(підпис)

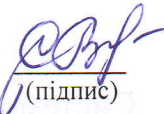
Черевко І.М.

(прізвище та ініціали)

Схвалено методичною радою факультету математики та інформатики.

Протокол № 11 від 25 червня 2024 року.

Голова методичної ради факультету

  
(підпис)

Сікора В.С.

(прізвище та ініціали)

©ЧНУ, 2024 рік

©Матвій О.В., 2024 рік

## **1. Мета навчальної дисципліни.**

Вступний курс для вивчення основ динамічної мови програмування Ruby з доступом до баз даних та основ теорії розподілених баз даних. Основними завданнями дисципліни «Теорія розробки розподілених баз даних» є отримання базових знань щодо розробки розподілених систем.

В рамках даної дисципліни студенти ознайомляться з основами мови програмування надвисокого рівня Ruby, особливо в частині динамічного програмування та використання багатопоточності, а також вдосконалять свої вміння та навички в області розробки розподілених систем, вивчаючи поняття та технічні рішення для забезпечення вичерпності, стійкості та зручного використання даних великих масштабів, а також різних протоколів та технологій, включаючи 2PC, 3PC, Saga, NoSQL, CAP.

**2. Результати навчання.** У результаті вивчення навчальної дисципліни студент повинен

### **знати:**

- основи мови програмування надвисокого рівня Ruby та її об'єктну модель;
- особливості використання мови Ruby та основи функціонального стилю програмування;
- інструменти динамічного програмування (створення коду, який сам себе може змінювати в процесі виконання) та багатопоточності;
- бібліотеки мови Ruby для роботи з базами даних;
- концепції та моделі розподілених систем;
- архітектури та технології для розробки розподілених систем (Мікросервісна архітектура, Message Queue, Service-Oriented Architecture);
- протоколи та методи комунікації між компонентами розподілених систем (REST, SOAP, gRPC);
- поняття та переваги різних дискретних типів даних (NoSQL, Relational databases);
- теорію CAP і її застосування в розробці розподілених систем;
- методи оптимізації та покращення ефективності роботи розподілених систем.

### **вміти:**

- використовувати синглтон класи, блоки, proc, lambda для організації налаштування програм, написаних засобами мови Ruby;
- використовувати функціональний стиль програмування з використанням блоків;
- застосовувати інструменти динамічного програмування мови Ruby для зміни коду в процесі його виконання;
- використовувати багатопоточність в програмах, написаних засобами мови Ruby;
- розробляти дизайн та архітектуру розподілених систем;
- реалізовувати різні протоколи та технології для розробки розподілених систем;
- реалізовувати мікросервісну архітектуру для розробки розподілених систем;

- працювати з різними типами даних та базами даних;
- визначати й вирішувати проблеми, що виникають при розробці та експлуатації розподілених систем;
- вдосконалювати та покращувати ефективність роботи розподілених систем;
- вивчати та реалізовувати алгоритми та структури даних для обробки великих обсягів даних.

Дисципліна формує такі **компетенції** у відповідності до стандарту вищої освіти спеціальності 124 – Системний аналіз та освітньої програми:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК3. Здатність вчитися та оволодівати сучасними знаннями.

ЗК4. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність розробляти проекти та управляти ними.

#### **фахові компетентності:**

СК2. Здатність проектувати архітектуру інформаційних систем.

СК10. Здатність до самоосвіти та професійного розвитку.

СК12. Здатність вивчати та критично оцінювати парадигми та нові методології розроблення моделі середовища розподілених систем баз даних та знань.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими **програмними результатами навчання**:

РН13. Здатність ефективно здійснювати вибір концептуальної моделі середовища розподілених систем баз даних та знань на основі структурного та об'єктно-орієнтованого підходів.

### **3. Опис навчальної дисципліни**

#### **3.1. Загальна інформація**

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	1	1	3	90	15	–	–	30	45	–	залік

### 3.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість кредитів - 3 / Кількість годин - 90 год												
	денна форма						Заочна форма						
	усього	у тому числі					усього	у тому числі					
		л	п	лаб	інд	с.р.		л	п	лаб	інд	ср	
1	2	3	4	5	6	7	8	9	10	11	12	13	
<b>Теми лекційних занять</b>	<b>Змістовий модуль 1. Реалізація мережевих інформаційних систем засобами Ruby</b>												
Основи мови програмування Ruby. Область застосування мови Ruby. Основні можливості. Базовий синтаксис. Базові класи: числа, стрічки, масиви, асоційовані масиви, множини, діапазони. Основи роботи з регулярними виразами. Використання блоків. Функціональний та імперативний стиль програмування у Ruby. Робота з блоками в Ruby на прикладі обробки файлів у ZIP-архівах з логуванням та управлінням помилками.	11	2		4		5							
Основи роботи з класами в Ruby. Об'єктна модель Ruby. Простір об'єктів. Система типів. Розподілене оголошення і перевизначення класів. Організація модулів та домішок. Відстрочені обчислення. Lazy-об'єкти. Використання Mixin Comparable та Enumerable у Ruby. Використання багатопоточності у Ruby. Приклад переведення документу з формату Markdown до HTML.	11	2		4		5							
Основи динамічнїюї програмування в Ruby, його переваги та недоліки. Основні концепції метапрограмування: код як дані, динамічне визначення методів, зміна класів і об'єктів, обробка відсутніх методів, використання рефлексії, динамічні зміни в контексті	11	2		4		5							

<p>класу та об'єкта, управління змінними. Тестування швидкодії коду при використанні різних підходів. Використання динамічного програмування на прикладі імітації створення HTTP запитів. Приклад створення адаптера для роботи з різними API з використанням метапрограмування. Створення Domain-Specific Languages (DSL) засобами Ruby.</p>												
<p>Використання бібліотек open-uri та Faraday для виконання HTTP-запитів різної складності. Основи використання бібліотек Nokogiri і Mechanize для парсингу веб-сторінок у Ruby. Формати даних CSV, JSON і YAML. Основи роботи з Selenium WebDriver. Основи роботи з вебсокетами у ruby. Приклад роботи веб-сервера з підтримкою вебсокетів. Використання бібліотеки Active Record для взаємодії з базою даних з використанням об'єктно-орієнтованого підходу.</p>	16	2	4		10							
<b>Разом за ЗМ1</b>	<b>49</b>	<b>8</b>	<b>16</b>		<b>25</b>							
<b>Теми лекційних занять</b>	<b>Змістовий модуль 2. Основи технології розробки розподілених систем</b>											
<p>Реляційні бази даних. Нормалізація. Транзакції. Принцип ACID.</p>	10	2	3		5							
<p>2PC та 3PC протоколи в розподілених системах. Паттерн Saga. Використання NoSQL. CAP теорема.</p>	10	2	3		5							
<p>Горизонтальне та вертикальне масштабування. Розподілення та синхронізація, консистентність даних. Мікросервісна</p>	21	3	8		10							

архітектура. Використання Kafka.											
<b>Разом за ЗМ 2</b>	<b>41</b>	<b>7</b>		<b>14</b>		<b>20</b>					
<b>Усього годин</b>	<b>90</b>	<b>15</b>		<b>30</b>		<b>45</b>					
<b>Підсумкова форма контролю</b>	залік (у першому семестрі)										

### 3.3. Тематика лабораторних занять

№	Назва теми	К-ть годин
1	Парсинг веб-даних: динамічне програмування та багатопоточність	16
2	Створення мікросервісної системи з використанням Kafka	14
	<b>ВСЬОГО</b>	<b>30</b>

### 3.4. Зміст завдань для самостійної роботи

Самостійна робота студентів складає 45 годин. Дана робота складається з підготовки до лекційних занять та повторення теоретичного матеріалу, самостійного опрацювання додаткового матеріалу, підготовки та виконання лабораторних занять та їх виконання, підготовки до заліку.

№ з/п	Назва теми	Кількість годин
1	Основи роботи з регулярними виразами. Використання блоків. Функціональний та імперативний стиль програмування у Ruby.	5
2	Об'єктна модель Ruby. Простір об'єктів. Організація модулів та домішок. Відстрочені обчислення. Lazy-об'єкти. Використання багатопоточності у Ruby.	5
3	Основи динамічне програмування в Ruby, його переваги та недоліки. Основні концепції метапрограмування.	5
4	Основи роботи з вебсокетами у ruby. Використання бібліотеки Active Record для взаємодії з базою даних з використанням об'єктно-орієнтованого підходу.	10
5	Реляційні бази даних. Нормалізація. Транзакції. Принцип ACID.	5
6	2PC та 3PC протоколи в розподілених системах. Паттерн Saga. Використання NoSQL. CAP теорема.	5
7	Мікросервісна архітектура. Використання Kafka.	10
	<b>Разом</b>	<b>45</b>

Студент може індивідуально виконувати додаткові завдання навчально-дослідницької спрямованості за завданнями, наданими викладачем.

Також студенти можуть отримати до 20 балів в рахунок ІНДЗ, якщо самостійно зареєструються на безкоштовних курсах платформи Prometheus або на аналогічних курсах платформи Coursera (за узгодженням з викладачем), пройдуть навчання, отримають відповідний сертифікат і надішлють його на сайт

дистанційного навчання викладачу разом зі скріншотом успішності на курсі. Кількість балів буде виставлена пропорційно до навчальних результатів студента (згідно зі статистикою сайту Prometheus чи Coursera).

#### **4. Освітні технології, методи навчання і викладання навчальної дисципліни**

Методи навчання:

- вербальні методи (лекція, бесіда, диспут, пояснення, розповідь тощо);
- практичні методи (лабораторні заняття);
- наочні методи (демонстрація, ілюстрація);
- робота з інформаційними ресурсами: з навчально-методичною, нормативною літературою та інтернет-ресурсами;
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни;
- дистанційне навчання з використанням відповідних онлайн-платформ (Moodle та Google Meet).

#### **5. Критерії та засоби оцінювання результатів навчання з навчальної дисципліни**

**5.1. Критерієм підсумкового оцінювання** є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним передбаченим результатом навчання навчальної дисципліни.

Перший модуль оцінюється в 40 балів, другий - 30 балів, підсумковий модуль - 30 балів.

Оцінювання першої лабораторної роботи здійснюється шляхом сумування чотирьох оцінок:

- використання принципів динамічного програмування - максимальна оцінка 10 балів;
- використання багатопоточності - максимальна оцінка 10 балів;
- використання об'єктно-орієнтованого підходу до реалізації проєкту - максимальна оцінка 10 балів;
- формування yaml файлів на основі отриманих даних, використання бібліотеки ActiveRecord та бази даних, бібліотеки для роботи з поштою - максимальна оцінка 10 балів.

Оцінювання другої лабораторної роботи здійснюється шляхом сумування трьох оцінок:

- написання контролера на стороні “споживача”, що буде приймати нові дані через REST запити, на надсилати дані в ProductService - максимальна оцінка 10 балів;
- створення відповідних сервісів в проєкті - максимальна оцінка 10 балів;
- додавання на стороні “споживача” код для збереження даних в SQL базі даних - максимальна оцінка 5 балів.

Максимальна кількість, яку можна набрати на підсумковому модулі (залік) – 30 балів.

Підсумкова оцінка виставляється за результатами суми балів набраних на змістовних модулях під час семестру та на підсумковому модулі (екзамені) згідно таблиці нижче.

### Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)		Кількість балів (залік)	Сумарна к-ть балів
Змістовий модуль №1	Змістовий модуль №2		
T1	T2	30	100
40	30		

### 5.2. Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
Зараховано	A (90-100)	відмінно
	B (80-89)	дуже добре
	C (70-79)	добре
	D (60-69)	задовільно
	E (50-59)	достатньо
Не зараховано	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим повторним курсом

На оцінку "відмінно" заслуговує студент, який виявив всебічні, систематичні та глибокі знання, здатність самостійно виконувати завдання, передбачені програмою, ознайомлений з основною і додатковою літературою, рекомендованою програмою. Така оцінка передбачає також засвоєння студентом взаємозв'язку основних понять дисципліни та їх значення для набутої професії.

Оцінку "добре" ставлять студентіві, який засвоїв навчально-програмовий матеріал у повному обсязі, успішно виконує передбачені програмою завдання, опрацював основну літературу, рекомендовану програмою, тобто студентіві, який засвідчив систематичний характер знань із дисципліни і здатний до їх самостійного поповнення й оновлення у процесі подальшої навчальної роботи і професійної діяльності.

На оцінку "задовільно" заслуговує студент, який виявив знання основного навчального матеріалу в обсязі, необхідному для подальшого навчання і майбутньої роботи за професією, здатний виконувати завдання, передбачені програмою, ознайомлений з основною літературою, рекомендованою програмою. Як правило, цю оцінку виставляють студентам, які припустилися

огріхів у відповіді на іспиті/заліку, але продемонстрували спроможність усунути їх.

Оцінку "незадовільно" ставлять студентів, у знаннях якого є прогалини, який припустився принципових помилок у виконанні передбачених програмою завдань, тобто студентів, який неспроможний продовжити навчання чи приступити до професійної діяльності після закінчення вищого навчального закладу без додаткових занять з відповідної дисципліни.

### **5.3. Засоби оцінювання**

Засобами оцінювання та демонстрування результатів навчання є:

- усний контроль у вигляді індивідуального та фронтального опитування на лекціях та лабораторних заняттях;
- захист лабораторних робіт та індивідуального навчально-дослідницького завдання;
- підсумкове тестове опитування.

### **6. Форми поточного та підсумкового контролю**

Формами поточного контролю є усна та/або письмова (тестування, лабораторна робота, ІНДЗ) відповідь студента.

Формою підсумкового контролю є залік.

### **7. Рекомендована література – основна**

1. Michael Hartl. Ruby on Rails Tutorial: Learn Web Development with Rails. Addison-Wesley Professional, 2020. 624 p.
2. Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson Education, 2020. 272 p.
3. Гайна Г. Основи проектування баз даних. Навч. посібник. Київ : Кондор, 2007. 208 с.
4. Ситник Н. Організація баз та сховищ даних. Київ, 2017. 148 с.
5. Campbell L. Database Reliability Engineering: Designing and Operating Resilient Database Systems. O'Reilly, 2017. 294 p.
6. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly, 2017. 590 p.
7. Newman S. Building Microservices: Designing Fine-Grained Systems. O'Reilly, 2021. 500 p.
8. Gwen Shapira Todd Palino Rajini Sivaram. Kafka: The Definitive Guide. Real-Time Data and Stream Processing at Scale. 2nd ed. O'Reilly, 2021. 426 p.

### **8. Інформаційні ресурси**

1. Доступ до курсу - <https://moodle.chnu.edu.ua/course/view.php?id=3862>.
2. Бібліотека ім. В.І. Вернадського. URL: <http://www.nbu.gov.ua/>
3. Нормативно-правова база України URL: <http://zakon3.rada.gov.ua/>
4. Державна служба статистики України URL: <http://www.ukrstat.gov.ua/>

## 9. Політика освітнього процесу

Здобувач зобов'язаний своєчасно та якісно виконувати всі отримані завдання; за необхідністю з метою з'ясування всіх незрозумілих під час самостійної та індивідуальної роботи питань, відвідувати консультації викладача.

Студенти мають дотримуватись правил академічної доброчесності відповідно до "Кодексу академічної доброчесності ЧНУ". Політика дотримання академічної доброчесності (відповідно до Закону України "Про вищу освіту") – викладання навчальної дисципліни ґрунтується на засадах академічної доброчесності – сукупності етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень. Наявність академічного плагіату в студентських роботах є підставою для виставлення негативної оцінки. Списування студентів під час проведення модульної контрольної роботи є підставою для дострокового припинення її складання та виставлення негативної оцінки.

Складання/перескладання заліків відбувається за встановленим деканатом розкладом.