МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіофізики та комп'ютерних технологій

Затверджено
На засіданні
кафедри радіофізики та комп'ютерних технологій
факультету електроніки та комп'ютерних технологій
Львівського національного університету імені Івана Франка
(протокол №15/23 від 29 серпня 2023 р.)

Завідувач кафедри:
_____ Іван КАРБОВНИК

Силабус з навчальної дисципліни "Архітектурні патерни високопродуктивних систем", що викладається в межах ОПП " Високопродуктивний комп'ютинг " першого (бакалаврського) рівня вищої освіти для здобувачів з спеціальності 121 – Інженерія програмного забезпечення

Львів 2023 р.

Назва дисципліни	Архітектурні патерни високопродуктивних систем
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79017
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіофізики та комп'ютерних технологій
Галузь знань, шифр та назва спеціальності	12 – Інформаційні технології, 121 – Інженерія програмного забезпечення
Викладачі дисципліни	Кушнір Олексій Олександрович, канд. фіз.-мат. наук, доцент
Контактна інформація викладачів	oleksiy.kushnir@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/kushnir-o-o
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі онлайн консультації через MS Teams. Для погодження часу онлайн консультацій слід писати на електронну пошту викладача.
Сторінка дисципліни	
Інформація про дисципліну	Дисципліна «Архітектурні патерни високопродуктивних систем» є вибірковою дисципліною з спеціальності 121 Інженерія програмного забезпечення для освітньої програми «Високопродуктивний комп'ютинг», яка викладається у 8 семестрі в обсязі 5,0 кредити (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб надати учасникам необхідні знання та навички для покращення роботи з шаблонами проєктування та побудови ефективної обчислюваної архітектури, що розрахована на обробку великих масивів даних та спроможна витримувати великі навантаження та стреси. У дисципліні представлено огляд засобів, технологій та концепцій, що використовуються у сучасній розробці програмних високопродуктивних застосунків та вимоги і стандарти, що для цього використовуються.
Мета та цілі дисципліни	Метою вивчення вибіркової дисципліни «Архітектурні патерни високопродуктивних систем» одержання студентами знань про шаблони та архітектурні патерни, які використовуються у сучасних високопродуктивних системах, отримати навички реалізації патернів та оптимізації системної архітектури.
Література для вивчення дисципліни	Основна література: 1. Будай А. Дизайн патерни – просто, як двері / А, Будай, 2012 – 90 с. 2. Ерік Фрімен. Патерни проектування. Фабула, 2020. – 672 с. 3. Sean Bradley. Design Patterns In Python. Common GoF (Gang of Four) Design Patterns Implemented In Python, 2020. – 238 p. 4. John F. Dooley. Software Development, Design and Coding. Apress, 2017. – 330 p. 5. Швець О. Refactoring and design patterns. Refactoring and Design Patterns. URL: https://refactoring.guru/. 6. Kasun Indrasiri. Design Patterns for Cloud Native Applications: Patterns in Practice Using APIs, Data, Events, and Streams. O'Reilly Media, 2020. – 314 с. 7. J Geewax. API Design Patterns. Manning Publications Co, 2020. -

	480 с. Додаткова література: 8. McConnell S. Code Complete, 2nd Edition, 2004, Cisco Press. 9. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship, 2008, Pearson
Обсяг курсу	Загальний обсяг: 150 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 86 год.
Очікувані результати навчання	Після завершення цього курсу студент буде: - Знати концепції та підходи побудови та оптимізації на всіх рівнях застосунків, розрахованих на велике обчислювальне навантаження. - Вміти розробляти та створювати високопродуктивні застосунки та оптимізовувати існуючі архітектури з оглядом на ефективність використання обчислюваних ресурсів за умов великого навантаження. Після вивчення даного курсу здобувачі набудуть таких Загальних(ЗК)/Фахових(ФК) компетентностей та Програмних результатів навчання (ПРН): ЗК01. Здатність до абстрактного мислення, аналізу та синтезу. ЗК02. Здатність застосовувати знання у практичних ситуаціях. ЗК05. Здатність вчитися і оволодівати сучасними знаннями. ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем. ФК16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами. ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу. ФК18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки). ФК19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення. ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення. ФК26. Здатність до алгоритмічного та логічного мислення. ФК28. Володіння методами розроблення систем підвищеної продуктивності, серверними та розподіленими технологіями, інструментальними засобами проектування та розробки веб-застосувань і нових технологій. ФК29. Здатність здійснювати розробку програмного забезпечення

	використовуючи різні методології та засоби програмування з метою забезпечення їх високої надійності та продуктивності в роботі. ПРН02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності. ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення. ПРН06. Вміння 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. ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення. ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення. 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. ПРН21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем. ПРН23. Вміти документувати та презентувати результати розробки програмного забезпечення. ПРН24. Вміти проводити розрахунок економічної ефективності програмних систем. ПРН26. Знати засоби інтеграції, розгортання та підтримки спеціалізованих програмних компонентів, розроблених на основі інноваційних технологій для вирішення завдань високопродуктивних обчислень.
Ключові слова	патерни, архітектурні шаблони, принципи проєктування, ефективність обчислень
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін «Програмування», «Об'єктно-орієнтоване програмування», «Архітектура ПЗ (ч. 1)», «Архітектура ПЗ (ч. 2)».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія, тестування, самооцінювання.

Необхідне обладнання	Мультимедіа, платформа Moodle, комп'ютерне програмне забезпечення Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), або краще, 8ГБ, або більше оперативної пам'яті, 50ГБ вільного місця на диску, вбудована або дискретна відеокарта), доступ до мережі Internet, засоби мультимедіа (в т.ч. проектор). Для проведення лабораторних занять: Комп'ютер (мінімальні характеристики: процесор з апаратною підтримкою віртуалізації Intel Core i3(4 ядра/8 потоків), або краще, 8ГБ, або більше оперативної пам'яті, 50ГБ вільного місця на диску, вбудована або дискретна відеокарта). Необхідне програмне забезпечення включає в себе ОС Windows/Linux/Mac, середовище розробки MS Visual Studio/PyCharm/Vim/VS Code/інше, компілятор мови програмування C++/Python.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • Змістовий модуль 1: 25% семестрової оцінки за тест модульного контролю. • Змістовий модуль 2: 25% семестрової оцінки за тест модульного контролю. • Виконання лабораторних робіт: 50% семестрової оцінки. Підсумкова максимальна кількість балів 100, звітність – залік в кінці семестру. _____. **Контрольні заміри проводяться у формі тестових завдань.** Модульний контроль проводиться за допомогою багаторівневих тестів: перший рівень – відповідь із запропонованих (1 правильна відповідь з 4), 2 рівень – відповідь цілим числом (номер рядку у якому помилка, або виконується певна дія), 3 рівень – відповідь словом, фразою, або числом, 4 рівень – есе із довільним текстом (програмний код або опис). **Академічна доброчесність**: Очікується, що лабораторні та контрольні роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахуванння викладачем, незалежно від масштабів плагіату чи обману. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом. **Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. **Політика виставлення балів.** Враховуються бали набрані на поточному тестуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на

	заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнень на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін. Жодні форми порушення академічної доброчесності не толеруються. **Оцінювання лабораторних робіт** (12 лабораторних робіт, максимальна кількість балів: 50) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії (0-5 балів за одну роботу) та захисту звіту по виконаній лабораторній роботі (0-5 балів за одну роботу). У підсумку, всі набрані бали множаться на коефіцієнт (0,417) для переведення у 50-а бальну шкалу. Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням: 5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання; 4 – студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання (або з несуттєвими недоліками); 3 – студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з помірними недоліками; 2 – студент погано розуміє розглянутий матеріал та написаний ним код програми, студент в більшості надає помилкові відповіді на питання по темі, код програми функціонує з суттєвими недоліками; 1 - студент погано розуміє розглянутий матеріал та написаний ним код програми, код програми не функціонує належним чином; 0 - студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі. **Оцінювання змістових модулів** (2 змістових модулі, 25 балів за кожний) — за результатами написаних студентом есе, тестів, програм, тощо. Бали оцінювання змістових модулів нараховуються за результатами тестування та нормується до 25 балів. **Критерії оцінювання результатів неформальної освіти:** Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних IT компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.
Питання до заліку чи екзамену	**Орієнтовний перелік питань та завдань для проведення підсумкової оцінки знань:**

	1. Що таке архітектурні патерни та яку роль вони відіграють у розробці програмного забезпечення високого навантаження? 2. Як архітектурний патерн MVC допомагає у розділенні компонентів у високопродуктивних системах? 3. Які приклади використання патерну Компонувальник у розробці програм? 4. Як працює Фабричний метод та в яких випадках його використання є доцільним? 5. Що таке патерн Одинак і як його можна використовувати для гарантованого існування лише одного екземпляра класу? 6. Як патерн Спостерігач допомагає реалізувати механізми підписки та сповіщення? 7. Як Ланцюжок обов'язків полегшує обробку запитів у великих та надвеликих системах? 8. Як архітектурні патерни впливають на розробку мікросервісних архітектур? 9. Які патерни оптимізації баз даних ви можете застосувати для великих об'ємів даних? 10. Які патерни проектування ефективно використовуються для асинхронних операцій у програмному забезпеченні? 11. Які патерни можуть бути використані для забезпечення високої безпеки та відмовостійкості системи? 12. Як можна інтегрувати сторонні сервіси у ваш додаток, використовуючи патерни та ефективні рішення? 13. Які стратегії розгортання та масштабування є ефективними для високопродуктивних систем? 14. Які архітектурні патерни можна використовувати для роботи з великими даними? 15. Як можна використовувати патерни кешування та оптимізації для обробки мільйонів одночасних запитів? 16. Які архітектурні патерни доцільно використовувати для розробки хмарно-орієнтованих високопродуктивних систем? 17. Які основні принципи лежать в основі паттернів проектування та чому вони важливі? 18. Як можна використовувати патерни для поліпшення інтерфейсу користувача та його взаємодії з системою? 19. Як патерни можуть бути застосовані для оптимізації роботи із великим даними у реальному часі?
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	**Вступ до Архітектурних Патернів** Огляд понять архітектурних патернів та їх роль у розробці високопродуктивних систем.	Лекція	1, 2, 3	2	1 тиж. семестру
1	Вступне заняття. Інструкція з техніки безпеки. Інсталяція та налаштування середовища розробки	Лабораторна робота	1, 2, 3	2	1 тиж. семестру
2	**Структурні патерни: модель-вид-контролер (MVC)** Розгляд патерну MVC і його використання для розділення компонентів у високопродуктивних системах. Підходи до оптимізації веб-застосунків.	Лекція	3	2	2 тиж. семестру
2	Простий веб-додаток, який використовує Ajax для асинхронного завантаження даних та оновлення інтерфейсу без перезавантаження сторінки	Лабораторна робота	1, 3	2	2 тиж. семестру
3	**Структурні патерни: компонувальник** Дослідження можливостей патерну Компонувальник для створення ієрархій об'єктів та його застосування у великих системах.	Лекція	2, 3	2	3 тиж. семестру
3	Автоматизація збору замовлень на складі з патерном Компонувальник	Лабораторна робота	2, 3	2	3 тиж. семестру
4	**Породжуючі патерни: фабричний метод** Огляд Фабричного методу та його використання для гнучкого створення об'єктів у високонавантажених системах.	Лекція	4, 5	2	4 тиж. семестру
4	Влясна реалізація Фабричного методу для поставленої задачі	Лабораторна робота	4, 5	2	4 тиж. семестру
5	**Породжуючі патерни: одинак** Розгляд патерну Одинак для гарантованого існування лише одного екземпляра класу у системі.	Лекція	5,7	2	5 тиж. семестру
5	Використання Одинака для створення єдиного логу	Лабораторна робота	3, 4, 5	2	5 тиж. семестру
6	**Поведінкові патерни: спостерігач** Використання патерну Спостерігач для реалізації механізмів підписки та сповіщення.	Лекція	2, 3	2	6 тиж. семестру

6	Реалізація моделі трекінгу посилок за допомогою патерну Спостерігач	Лабораторна робота	1, 2, 3, 4	2	6 тиж. семестру
7	**Поведінкові Патерни: Ланцюжок обов'язків** Огляд патерну ланцюжок обов'язків для покращення обробки запитів у великих системах.	Лекція	2, 3	2	7 тиж. семестру
7	Підсумкове заняття ЗМ 1	Лабораторна робота	1, 5	2	7 тиж. семестру
8	**Архітектурні Патерни в мікросервісах** Розгляд використання архітектурних патернів у розробці мікросервісних архітектур.	Лекція	1, 5,7	2	8 тиж. семестру
8	Мікросервісний застосунок, що складається з невеликих сервісів, які взаємодіють між собою через API.	Лабораторна робота	1, 2, 3, 4, 5	2	8 тиж. семестру
9	**Патерни для оптимізації баз даних** Огляд патернів, спрямованих на оптимізацію роботи з базами даних у високонавантажених системах.	Лекція	1,6	4	9 тиж. семестру
9	Оптимізація запитів до бази даних у заданому проекті, використовуючи індексацію та оптимізацію SQL-запитів.	Лабораторна робота	1	2	9 тиж. семестру
10	**Шаблони проектування для асинхронних операцій** Огляд патернів проектування для ефективної роботи з асинхронними операціями.	Лекція	1, 2, 4	2	10 тиж. семестру
10	Викоримстання патернів Promise та async/await для оптимізації викликів асинхронних функцій.	Лабораторна робота	1, 3	2	10 тиж. семестру
11	**Патерни безпеки високонавантажених систем** Розгляд патернів, спрямованих на забезпечення безпеки та відмовостійкості системи.	Лабораторна робота	1, 2, 4	2	11 тиж. семестру
11	Створення захищеного механізму автентифікації та авторизації для високонавантаженої системи.		1, 3, 5, 6	2	11 тиж. семестру
12	**Інтеграція із сервісами третьої сторони: Патерни та рішення** Пошук архітектурних патернів для успішної інтеграції з сервісами третьої сторони.	Лекція	1, 3, 4	2	12 тиж. семестру
12	Інтеграція Google Maps API у власний застосунок для відображення даних на мапі	Лабораторна робота	1, 3, 4	2	12 тиж. семестру
13	**Патерни розгортання та масштабування** Вивчення ефективних стратегій розгортання та масштабування високопродуктивних систем.	Лекція	1, 2	2	13 тиж. семестру

13	Використання Docker, для масштабування застосунку	Лабораторна робота	1, 2	2	13 тиж. семестру
14	**Патерни для роботи з Big Data** Огляд патернів, спрямованих на оптимізацію та ефективну роботу з великими обсягами даних.	Лекція	1, 2, 3	2	14 тиж. семестру
15	**Патерни для роботи з мільйонами запитів: кешування та оптимізація** Дослідження методів кешування та оптимізації для роботи з великим обсягом одночасних запитів.	Лекція	1, 2, 3	2	15 тиж. семестру
14-15	Застосування кешування для оптимізації відповідей сервера на мільйони одночасних запитів	Лабораторна робота	1, 2, 3	4	15 тиж. семестру
16	**Архітектурні патерни для Cloud-Native розробки** Огляд архітектурних патернів, які підходять для розробки хмарно-орієнтованих високопродуктивних систем.	Лекція	1, 2, 6	2	16 тиж. семестру
16	Підсумкове заняття ЗМ 2	Лабораторна робота	1, 2, 3, 4, 5	2	16 тиж. семестру