МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра обчислювальної математики

Затверджено
На засіданні
кафедри обчислювальної математики
факультету прикладної математики та інформатики
Львівського національного університету імені Івана Франка
(протокол № 1 від 29 серпня 2023 р.)

Завідувач кафедри
_____ Роман ХАПКО

Силабус з навчальної дисципліни "Інтеграція програмних систем", що викладається в межах ОПП Прикладна математика другого (магістерського) рівня вищої освіти для здобувачів з спеціальності 113 – прикладна математика

Львів 2023 р.

Назва дисципліни	Інтеграція програмних систем
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра обчислювальної математики
Галузь знань, шифр та назва спеціальності	11 – математика і статистика 113 – прикладна математика
Викладачі дисципліни	Вавричук Василь Григорович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики Музичук Юрій Анатолійович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики
Контактна інформація викладачів	vasyl.vavrychuk@lnu.edu.ua, https://ami.lnu.edu.ua/employee/vavrychuk yuriy.muzychuk@lnu.edu.ua, https://ami.lnu.edu.ua/employee/muzychuk-yuriy Головний корпус ЛНУ ім. І. Франка, каб. 360. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять, за розкладом консультацій кафедри, а також в середовищі Microsoft Teams.
Сторінка курсу	https://ami.lnu.edu.ua/course/intehratsiia-prohramnykh-system-prykladna-matematyka
Інформація про дисципліну	Дисципліна "Інтеграція програмних систем" є вибірковою дисципліною для спеціальності 113 - прикладна математика для освітньої програми "Прикладна математика", яка викладається в 2-му семестрі в обсязі 4.5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Розвиток IT індустрії та програмування на даний час дозволяє ефективно вирішувати широке коло задач, які ставляться замовниками: виробництвом, бізнесом, державним сектором і т.д. Великою мірою це забезпечується завдяки поширенню розподілених архітектурних рішень (багатошарова архітектура, гексагональна архітектура, мікросервіси і т.д.), перевикористанню різноманітних технологій і фреймворків (фронтенд, бекенд, комунікація і т.д.), та запровадженню ефективних практик (SCM, CI і т.д.). У циклі цього курсу студенти знайомляться з фрейморками та бібліотеками (керування браузером, черги, GraphQL і т.д.), які доступні під різні мови програмування, в залежності від того чому розробник надає перевагу. Розгалядаються поширені технології такі, як черги, GraphQL, JWT та Docker, які забезпечують розподіленість архітектури. У випадку інтерфейсу користувача, фреймворки, що розгядаються є, на жаль, мово-залежними: JavaFX - під мову Java, та Angular - під мову TypeScript або JavaScript. Проте, багато принципів, на яких побудований JavaFX мають місце і для графічних фреймворків під інші мови програмування, наприклад WPF під C#. У випадку Angular, він порівнюється із іншим поширеним фронтенд фреймворком React. У курсі засоби, принципи та підходи розглядаються з практичною метою: студенти реалізовують проект, що можна назвати "золотим стандартом" IT проекту: програма, що надає можливості CRUD операцій з даними та графічний інтерфейс. Для легкого забезпечення більшої реалісничності завдання, студентам

	необхідно отримати дані з реальних джерел, таких як сайти новин, інтернет магазини і т.д.
Мета та цілі дисципліни	Метою навчальної дисципліни є розширення навичок студентів роботи з фреймворками та бібліотеками під мови програмування, на яких вони спеціалізуються. Набуття знань з побудови, реалізації та інтеграції розподілених архітектурних рішень. Здобуття вмінь з ефективних практик з розробки програмного забезпечення (SCM, неперервна інтеграція і т.д.). Здобуття практичного досвіду підготовки стандартного проекту з IT, що включає поширений набір функціональності та є достатньо складним.
Література для вивчення дисципліни	1. https://www.selenium.dev 2. https://playwright.dev 3. RabbitMQ Tutorials 4. GraphQL > Learn 5. GraphQL .NET > Docs 6. https://jwt.io 7. https://docs.docker.com 8. https://angular.io 9. Підручник з JavaFX 10. https://plantuml.com
Обсяг курсу	Загальний обсяг: 135 годин. Аудиторних занять: 48 год., з них 32 год. лекцій та 16 год. лабораторних робіт. Самостійної роботи: 87 год.
Очікувані результати навчання	Після завершення цього курсу студент буде знати: • можливості нових фреймворків та бібліотек • ефективні практики з розробки програмного забезпечення вміти: • працювати з новими фреймворками та бібліотеками • об'єднувати декілька компонент в цілісну систему • реалізовувати стандартний проект з IT • застосовувати ефективні практики з розробки програмного забезпечення
Ключові слова	IT, програмування, прикладна математика
Формат курсу	Очний
Теми	Подано нижче у таблиці «Схема курсу»
Підсумковий контроль, форма	Залік у кінці семестру.
Пререквізити	Для вивчення курсу студенти потребують базових знань з програмування, баз даних, веб-розробки, операційних систем.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентації, лекції, модульний контроль, індивідуальні завдання.
Необхідне обладнання	Комп'ютер, Internet.
Критерії оцінювання (окремо для кожного виду навчальної	Під час семестру студент може отримати 100 балів: по 20 балів за 5 індивідуальних завдань. **Академічна доброчесність**: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на

діяльності)	використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів письмових робіт та індивідуальних завдань, 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 за виконання індивідуальних завдань. 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.
Приклади питання при захисті індивідуальних завдань.	1. Навести приклади використання селекторів для отримання даних з веб-сторінки. 2. Чи використовуються очікування або затримки при отриманні даних з веб-сторінки? Чому вони мають використовуватися, або чому вони можуть не використовуватися? 3. Пояснити алгоритм пагінації по сторінках веб-сайту при отриманні даних з нього. 4. Модифікувати програму, щоб отримати додаткові дані з веб-сторінки. 5. Модифікувати програму, щоб пересилати додаткові дані через чергу. 6. Продемонструвати використання черги через засоби моніторингу черги. 7. Пояснити основні етапи реалізації GraphQL сервера засобами фреймворку. 8. Виконати типові запити до GraphQL сервера. 9. Пояснити шлях опрацювання GraphQL запиту. 10. Модифікувати програму, щоб повертати додаткові дані через GraphQL сервер. 11. Модифікувати програму, щоб реалізувати додаткові GraphQL мутації. 12. Пояснити основні етапи реалізації JWT автентифікації. 13. Пояснити послідовність JWT автентифікації. 14. Основні концепції Docker: образ, контейнер, CLI, Dockerfile. 15. Замінити один з ваших контейнерів процесом, що виконується на хост-машині. 16. Додати контейнер, який буде виконувати запит до одного із сервісів. 17. Пояснити основні етапи реалізації UI засобами фреймворку. 18. Модифікувати програму, щоб відображати в UI додаткові дані. 19. Основні види UML діаграм: діаграми класів, послідовностей,

	розгортання, машин станів.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

Схема курсу “Інтеграція програмних систем”

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
1	**Тема 1. Вступ** Огляд компонент індивідуального завдання.	Лекція (2 год.)	[1-10]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 1. Вступ Налаштування робочого середовища, WSL. Індивідуальне завдання №1: Засоби керування браузером та черги.	Лабораторна робота (2 год.)	https://learn.microsoft.com/en-us/windows/wsl/	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
2	**Тема 2. Інструмент керування браузером Selenium WebDriver** Інсталяція, драйвери, селектори, затримки.	Лекція (2 год.)	[1]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №1 (3 год.)	1 тиждень
3	**Тема 3. Інструмент керування браузером playwright** Інсталяція, дії, перевірки.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №1 (3 год.)	1 тиждень
	Тема 2-3. Інструменти керування браузером WebDriver та playwright Приклади отримання даних з веб-сторінок використовуючи засоби керування браузером. Проходження по сторінках з пагінацією.	Лабораторна робота (2 год.)		Виконання індивідуального завдання №1 (3 год.)	1 тиждень
4	**Тема 4. Черги: RabbitMQ, Redis, і т.д.** Найпростіший приклад, черги для розпаралелювання, publish/subscribe. роутінг, топіки, RPC.	Лекція (2 год.)	[3]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №1 (3 год.)	1 тиждень
5	**Тема 5. Мова запитів GraphQL** Структура запиту, мутації, поля,	Лекція (2 год.)	[4]	Опрацювання лекційного матеріалу, виконання	1 тиждень

	аргументи, аліаси, фрагменти, ім’я операції, змінні, директиви, інлайн фрагменти.			індивідуальн ого завдання №1 (4 год.)	
	Тема 2-4. Засоби керування браузером та черги Здача індивідуального завдання №1. Індивідуальне завдання №2: .	Лабораторн на робота (2 год.)		Виконання індивідуальн ого завдання №2 (3 год.)	1 тиждень
6	**Тема 6. Мова типів GraphQL** Об’єкти, поля, аргументи, скалярні типи, перелічування, списки, інтерфейс, не нульові типи, об’єднання, типи введення.	Лекція (2 год.)	[4]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №2 (4 год.)	1 тиждень
7	**Тема 7. Фреймворки реалізації GraphQL сервера** Інсталяція, схемоцентрований підхід, GraphQL типоцентрований підхід.	Лекція (2 год.)	[5]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №2 (4 год.)	1 тиждень
	Тема 5-7. GraphQL Здача індивідуального завдання №2. Індивідуальне завдання №3: Автентифікація JWT.	Лабораторн на робота (2 год.)		Виконання індивідуальн ого завдання №3 (3 год.)	1 тиждень
8	**Тема 8. Автентифікація JWT** Структура токена, створення, перевірка токена.	Лекція (2 год.)	[6]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №3 (4 год.)	1 тиждень
9	**Тема 9. Вступ у Docker** Образ, контейнер, CLI, Dockerfile.	Лекція (2 год.)	[7]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №3 (4 год.)	1 тиждень
	Тема 9. Вступ у Docker Запуск веб-аплікації, бази даних в контейнері.	Лабораторн на робота (2 год.)		Виконання індивідуальн ого завдання №3 (3 год.)	1 тиждень
10	**Тема 10. Docker compose**	Лекція (2 год.)	[7]	Опрацювання лекційного	1 тиждень

	Заміна запуску декількох контейнерів на Docker compose.			матеріалу, виконання індивідуальн ого завдання №3 (4 год.)	
11	**Тема 11. Devcontainers** Devcontainer-ри у VS Code. Проблема UID та GID.	Лекція (2 год.)	[7], https://containers.dev, https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №3 (4 год.)	1 тиждень
	Тема 8. JWT Здача індивідуального завдання №3. Індивідуальне завдання №4: Контейнери.	Лабораторна робота (2 год.)		Виконання індивідуальн ого завдання №4 (3 год.)	1 тиждень
12	**Тема 12. Мова TypeScript** Node.js, вступ у TS, порівняння з іншими мовами, основи, типи, звуження, класи, модулі.	Лекція (2 год.)	https://www.typescriptlang.org	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №4 (4 год.)	1 тиждень
13	**Тема 13. Фронтенд фреймворк Angular** Створення аплікації, додавання CRUD можливостей, взаємодія між компонентами.	Лекція (2 год.)	[8]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №4 (4 год.)	1 тиждень
	Тема 9-11. Контейнери Здача індивідуального завдання №4. Індивідуальне завдання №5: Фронтенд.	Лабораторна робота (2 год.)		Виконання індивідуальн ого завдання №5 (3 год.)	1 тиждень
14	**Тема 14. Реактивні засоби інтерфейсу користувача JavaFX** Scene Builder, модель та	Лекція (2 год.)	[9]	Опрацювання лекційного матеріалу, виконання індивідуальн	1 тиждень

	компонент TableView, взаємодія з користувачем.			ого завдання №5 (4 год.)	
15	**Тема 15. UML, PlantUML** Діаграми класів, послідовностей, розгортання, машин станів.	Лекція (2 год.)	[10]	Опрацювання лекційного матеріалу, виконання індивідуальн ого завдання №5 (4 год.)	1 тиждень
	Тема 9-11. Фронтенд Здача індивідуального завдання №5.	Лабораторн на робота (2 год.)			1 тиждень
16	**Тема 16. Підсумкова лекція** Обговорення поширених проблем та альтернативних методів реалізації індивідуального завдання.	Лекція (2 год.)	[1-10]		1 тиждень