

**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.

<b>Назва дисципліни</b>	Інтеграція програмних систем
<b>Адреса викладання дисципліни</b>	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики Кафедра обчислювальної математики
<b>Галузь знань, шифр та назва спеціальності</b>	11 – математика і статистика 113 – прикладна математика
<b>Викладачі дисципліни</b>	Вавричук Василь Григорович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики Музичук Юрій Анатолійович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики
<b>Контактна інформація викладачів</b>	vasyl.vavrychuk@lnu.edu.ua, <a href="https://ami.lnu.edu.ua/employee/vavrychuk">https://ami.lnu.edu.ua/employee/vavrychuk</a> yuriy.muzychuk@lnu.edu.ua, <a href="https://ami.lnu.edu.ua/employee/muzychuk-yuriy">https://ami.lnu.edu.ua/employee/muzychuk-yuriy</a> Головний корпус ЛНУ ім. І. Франка, каб. 360. м. Львів, вул. Університетська, 1
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять, за розкладом консультацій кафедри, а також в середовищі Microsoft Teams.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/intehratsiia-prohramnykh-system-prykladna-matematyka">https://ami.lnu.edu.ua/course/intehratsiia-prohramnykh-system-prykladna-matematyka</a>
<b>Інформація про дисципліну</b>	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).
<b>Коротка анотація дисципліни</b>	Розвиток ІТ 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 фреймворками та бібліотеками (керування браузером, черги, GraphQL і т.д.), які доступні під різні мови програмування, в залежності від того чому розробник надає перевагу. Розглядаються поширені технології такі, як черги, GraphQL, JWT та Docker, які забезпечують розподіленість архітектури. У випадку інтерфейсу користувача, фреймворки, що розглядаються є, на жаль, мовозалежними: 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 реалізують проект, що можна назвати "золотим стандартом" ІТ проекту: програма, що надає можливості CRUD операцій з даними та графічний інтерфейс. Для легкого забезпечення більшої реалістичності завдання, студентам

	необхідно отримати дані з реальних джерел, таких як сайти новин, інтернет магазини і т.д.
<b>Мета та цілі дисципліни</b>	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 ІТ, що включає поширений набір функціональності та є достатньо складним.
<b>Література для вивчення дисципліни</b>	<ol style="list-style-type: none"> <li>1. <a href="https://www.selenium.dev">https://www.selenium.dev</a></li> <li>2. <a href="https://playwright.dev">https://playwright.dev</a></li> <li>3. <a href="#">RabbitMQ Tutorials</a></li> <li>4. <a href="#">GraphQL &gt; Learn</a></li> <li>5. <a href="#">GraphQL .NET &gt; Docs</a></li> <li>6. <a href="https://jwt.io">https://jwt.io</a></li> <li>7. <a href="https://docs.docker.com">https://docs.docker.com</a></li> <li>8. <a href="https://angular.io">https://angular.io</a></li> <li>9. Підручник з JavaFX</li> <li>10. <a href="https://plantuml.com">https://plantuml.com</a></li> </ol>
<b>Обсяг курсу</b>	Загальний обсяг: 135 годин. Аудиторних занять: 48 год., з них 32 год. лекцій та 16 год. лабораторних робіт. Самостійної роботи: 87 год.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде знати:</p> <ul style="list-style-type: none"> <li>• можливості нових фреймворків та бібліотек</li> <li>• ефективні практики з розробки програмного забезпечення</li> </ul> <p>вміти:</p> <ul style="list-style-type: none"> <li>• працювати з новими фреймворками та бібліотеками</li> <li>• об'єднувати декілька компонент в цілісну систему</li> <li>• реалізовувати стандартний проект з ІТ</li> <li>• застосовувати ефективні практики з розробки програмного забезпечення</li> </ul>
<b>Ключові слова</b>	ІТ, програмування, прикладна математика
<b>Формат курсу</b>	Очний
<b>Теми</b>	Подано нижче у таблиці «Схема курсу»
<b>Підсумковий контроль, форма</b>	Залік у кінці семестру.
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з програмування, баз даних, веб-розробки, операційних систем.
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Презентації, лекції, модульний контроль, індивідуальні завдання.
<b>Необхідне обладнання</b>	Комп'ютер, Internet.
<b>Критерії оцінювання (окремо для кожного виду навчальної)</b>	<p>Під час семестру студент може отримати 100 балів: по 20 балів за 5 індивідуальних завдань.</p> <p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на</p>

<p>діяльності)</p>	<p>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.</p> <p><b>Відвідання занять</b> 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.</p> <p><b>Література.</b> 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.</p> <p><b>Політика виставлення балів.</b> 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 запізнь 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.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
<p><b>Приклади питання при захисті індивідуальних завдань.</b></p>	<ol style="list-style-type: none"> <li>1. Навести приклади використання селекторів для отримання даних з веб-сторінки.</li> <li>2. Чи використовуються очікування або затримки при отриманні даних з веб-сторінки? Чому вони мають використовуватися, або чому вони можуть не використовуватися?</li> <li>3. Пояснити алгоритм пагінації по сторінках веб-сайту при отриманні даних з нього.</li> <li>4. Модифікувати програму, щоб отримати додаткові дані з веб-сторінки.</li> <li>5. Модифікувати програму, щоб пересилати додаткові дані через чергу.</li> <li>6. Продемонструвати використання черги через засоби моніторингу черги.</li> <li>7. Пояснити основні етапи реалізації GraphQL сервера засобами фреймворку.</li> <li>8. Виконати типові запити до GraphQL сервера.</li> <li>9. Пояснити шлях опрацювання GraphQL запиту.</li> <li>10. Модифікувати програму, щоб повертати додаткові дані через GraphQL сервер.</li> <li>11. Модифікувати програму, щоб реалізувати додаткові GraphQL мутації.</li> <li>12. Пояснити основні етапи реалізації JWT автентифікації.</li> <li>13. Пояснити послідовність JWT автентифікації.</li> <li>14. Основні концепції Docker: образ, контейнер, CLI, Dockerfile.</li> <li>15. Замінити один з ваших контейнерів процесом, що виконується на хост-машині.</li> <li>16. Додати контейнер, який буде виконувати запит до одного із сервісів.</li> <li>17. Пояснити основні етапи реалізації UI засобами фреймворку.</li> <li>18. Модифікувати програму, щоб відображати в UI додаткові дані.</li> <li>19. Основні види UML діаграм: діаграми класів, послідовностей,</li> </ol>

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

Схема курсу “Інтеграція програмних систем”

Ти ж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
1	<b>Тема 1. Вступ</b> Огляд компонент індивідуального завдання.	Лекція (2 год.)	[1-10]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	<b>Тема 1. Вступ</b> Налаштування робочого середовища, WSL. Індивідуальне завдання №1: Засоби керування браузером та черги.	Лабораторна робота (2 год.)	<a href="https://learn.microsoft.com/en-us/windows/wsl/">https://learn.microsoft.com/en-us/windows/wsl/</a>	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
2	<b>Тема 2. Інструмент керування браузером Selenium WebDriver</b>  Інсталяція, драйвери, селектори, затримки.	Лекція (2 год.)	[1]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №1 (3 год.)	1 тиждень
3	<b>Тема 3. Інструмент керування браузером playwright</b>  Інсталяція, дії, перевірки.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №1 (3 год.)	1 тиждень
	<b>Тема 2-3. Інструменти керування браузером WebDriver та playwright</b>  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	<b>Тема 4. Черги: RabbitMQ, Redis, і т.д.</b>  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	<b>Тема 5. Мова запитів GraphQL</b>  Структура запиту, мутації, поля,	Лекція (2 год.)	[4]	Опрацювання лекційного матеріалу, виконання	1 тиждень

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

	Заміна запуску декількох контейнерів на Docker compose.			матеріалу, виконання індивідуального завдання №3 (4 год.)	
11	<b>Тема 11. Devcontainers</b> Devcontainer-ри у VS Code. Проблема UID та GID.	Лекція (2 год.)	[7], <a href="https://containers.dev">https://containers.dev</a> , <a href="https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration">https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration</a>	Опрацювання лекційного матеріалу, виконання індивідуального завдання №3 (4 год.)	1 тиждень
	<b>Тема 8. JWT</b> Здача індивідуального завдання №3. Індивідуальне завдання №4: Контейнери.	Лабораторна робота (2 год.)		Виконання індивідуального завдання №4 (3 год.)	1 тиждень
12	<b>Тема 12. Мова TypeScript</b> Node.js, вступ у TS, порівняння з іншими мовами, основи, типи, звуження, класи, модулі.	Лекція (2 год.)	<a href="https://www.typescriptlang.org">https://www.typescriptlang.org</a>	Опрацювання лекційного матеріалу, виконання індивідуального завдання №4 (4 год.)	1 тиждень
13	<b>Тема 13. Фронтенд фреймворк Angular</b> Створення аплікації, додавання CRUD можливостей, взаємодія між компонентами.	Лекція (2 год.)	[8]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №4 (4 год.)	1 тиждень
	<b>Тема 9-11. Контейнери</b> Здача індивідуального завдання №4. Індивідуальне завдання №5: Фронтенд.	Лабораторна робота (2 год.)		Виконання індивідуального завдання №5 (3 год.)	1 тиждень
14	<b>Тема 14. Реактивні засоби інтерфейсу користувача JavaFX</b> 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	<b>Тема 15. UML, PlantUML</b> Діаграми класів, послідовностей, розгортання, машин станів.	Лекція (2 год.)	[10]	Опрацювання лекційного матеріалу, виконання індивідуального завдання №5 (4 год.)	1 тиждень
	<b>Тема 9-11. Фронтенд</b> Здача індивідуального завдання №5.	Лабораторна робота (2 год.)			1 тиждень
16	<b>Тема 16. Підсумкова лекція</b> Обговорення поширених проблем та альтернативних методів реалізації індивідуального завдання.	Лекція (2 год.)	[1-10]		1 тиждень