

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра радіоелектронних і комп'ютерних систем**

**Затверджено**

На засіданні кафедри РКС  
факультету електроніки та комп'ютерних  
технологій  
Львівського національного університету  
імені Івана Франка  
(протокол № 1/23 від 31.08.2022 р.)

Завідувач кафедри

 Ігор ОЛЕНИЧ

**Силабус з навчальної дисципліни**  
**«Архітектура програмного забезпечення»,**  
**що викладається в межах ОПП «Інженерія програмного**  
**забезпечення» першого (бакалаврського) рівня вищої освіти для**  
**здобувачів з спеціальності**  
**121 «Інженерія програмного забезпечення»**

<b>Назва дисципліни</b>	Архітектура програмного забезпечення
<b>Адреса викладання дисципліни</b>	м. Львів, вул. Драгоманова, 50
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій, кафедра радіоелектронних і комп'ютерних систем
<b>Галузь знань, шифр та назва спеціальності</b>	12 Інформаційні технології, 121 Інженерія програмного забезпечення
<b>Викладачі дисципліни</b>	Оленич Ігор Богданович, д-р фіз.-мат. наук, професор, завідувач кафедри Сінкевич Олег Олександрович, асистент
<b>Контактна інформація викладачів</b>	igor.olenych@lnu.edu.ua, <a href="https://electronics.lnu.edu.ua/employee/olenych-i-b">https://electronics.lnu.edu.ua/employee/olenych-i-b</a> oleh.sinkevych@lnu.edu.ua, <a href="https://electronics.lnu.edu.ua/employee/o_sinkevych">https://electronics.lnu.edu.ua/employee/o_sinkevych</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
<b>Сторінка дисципліни</b>	<a href="http://194.44.208.156/moodle/course/view.php?id=138">http://194.44.208.156/moodle/course/view.php?id=138</a>
<b>Інформація про дисципліну</b>	Дисципліна «Архітектура програмного забезпечення» є нормативною дисципліною з спеціальності 121 Інженерія програмного забезпечення для освітньої програми «Інженерія програмного забезпечення», яка викладається у 6 семестрі в обсязі 4,5 кредити (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	Навчальну дисципліну розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб брати участь у проектуванні програмного забезпечення, створенні інтерфейсів користувача та управлінні структурою програмного забезпечення. Тому у дисципліні представлено як огляд сучасних підходів створення складних програмних систем, так і засобів та інструментів, які потрібні для проектування програмного забезпечення, реалізації основних методів і засобів автоматизації проектування, випробувань та оцінки якості програмного забезпечення.
<b>Мета та цілі дисципліни</b>	<i>Мета:</i> надати студентам основні поняття про структуру, інформаційні моделі та системи побудови програмного забезпечення. <i>Цілі:</i> ознайомлення студентів з основними підходами, інформаційними моделями та системами побудови програмного забезпечення, мовами програмування та проектування програмного забезпечення.
<b>Література для вивчення дисципліни</b>	Основна література: 1. Martin R.C. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin. - Pearson, 2017. - 432 p. 2. Clements P., Bachmann F., Bass L. Documenting Software Architectures: Views and Beyond (2nd Edition). - Boston: Addison-Wesley Professional, 2010. 3. Goma H. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures. – NY: Cambridge University Press, 2011. – 578 p. 4. James Gough. Mastering API Architecture: Design, Operate, and Evolve API-Based Systems / J. Gough, D. Bryant, M. Auburn. O'Reilly Media; 1st edition, 2022. - 286 p.  Додаткова література: 5. Bewis T. C# Design Pattern Essentials. – NY: Ability First Limited, 2012. – 264 p.

	<p>6. Maciaszek L. Requirements analysis and systems design / L. Maciaszek. – Ontario: Pearson Education Canada, 2007. - 656 p.</p> <p>7. Richards M. Fundamentals of Software Architecture. An Engineering Approach / M. Richards, N. Ford. - Sebastopol, CA: O'Reilly Media, 2020. - 422 p.</p> <p>8. Brown S. Visualise, document and explore your software architecture. Software Architecture for Developers / Simon Brown. - Leanpub, 2019.</p> <p>9. Eyskens S. Software Architecture for Busy Developers / S. Eyskens. - Birmingham: Packt Publishing, 2021. - 174 p.</p>
<p><b>Обсяг курсу</b></p>	<p>64 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт та 71 година самостійної роботи</p>
<p><b>Очікувані результати навчання</b></p>	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> <li>- <i>Знати</i> методологію та технологію збору основних вимог до програмного забезпечення та методи їх аналізу; технології проектування програмного забезпечення; способи використання CASE засобів для аналізу предметного середовища та побудови архітектури програмного забезпечення.</li> <li>- <i>Вміти</i> виконувати аналіз вимог до програмного забезпечення, що розробляється; оцінювати трудомісткість і вибрати адекватні підходи до розробки програмного забезпечення; проектувати архітектуру програмного забезпечення з використанням засобів візуального моделювання; усвідомлено застосовувати методики випробувань і налагодження розроблюваного програмного забезпечення; проектувати компоненти архітектурного рішення.</li> </ul> <p>У результаті вивчення курсу «Архітектура програмного забезпечення» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК28. Володіння методами сучасних веб-технологій, хмарних технологій, великих даних та засобами розробки веб-застосувань.</p> <p>ПРН 1. Аналізувати, цілеспрямовано шукати і вибрати необхідні для вирішення професійних завдань інформаційно-довідкові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПРН6. Уміння вибрати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПРН11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання</p>

	<p>програмного забезпечення та структур даних і знань.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.</p> <p>ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПРН27. Вміти обирати оптимальні алгоритми та технології розробки програмного забезпечення.</p>
<b>Ключові слова</b>	Бізнес-логіка, структура програмного забезпечення, дизайн рівнів програмного забезпечення, патерни, CASE засоби
<b>Формат курсу</b>	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем
<b>Теми</b>	Див. СХЕМА КУРСУ
<b>Підсумковий контроль, форма</b>	Екзамен у формі захисту командних проєктів у кінці семестру
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін «Вступ в інженерію програмного забезпечення», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування», «Функційне програмування», «Патерни проектування».
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
<b>Необхідне обладнання</b>	Мультимедіа, платформа Moodle, комп'ютерне програмне забезпечення
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням:</p> <ul style="list-style-type: none"> <li>• лабораторні роботи: 25% семестрової оцінки; максимальна кількість балів 25.</li> <li>• контрольні заміри (2 модулі): 25% семестрової оцінки; максимальна кількість балів 25.</li> <li>• екзамен: 50% семестрової оцінки; максимальна кількість балів 50.</li> </ul> <p>Загалом упродовж семестру 100 балів.</p> <hr/> <p><b>Контрольні заміри проводяться у формі тестових завдань.</b></p> <p><b>Академічна доброчесність:</b> 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 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 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.</p> <p><b>Література.</b> Уся література, яку студенти не зможуть знайти самотійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до</p>

	використання також й іншої літератури та джерел, яких немає серед рекомендованих. <b>Політика виставлення балів.</b> 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 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. Жодні форми порушення академічної доброчесності не толеруються.
<b>Питання до контрольних робіт</b>	Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт розміщені на веб-сторінці курсу.
<b>Опитування</b>	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	<b>Основні відомості про архітектуру ПЗ.</b> Особливості розробки великих програмних систем. Архітектура і дизайн. Мета архітектури. Функціональність та структура ПЗ. Роль архітектури та архітектора.	Лекція	1, 2, 4, 9	Вступне заняття. Формування команд, обрання теми проєкту.	1 тиж. семестру
2	<b>Парадигми програмування.</b> Структурне програмування. Об'єктно-орієнтоване програмування. Інкапсуляція, успадкування, поліморфізм. Функційне програмування.	Лекція	1, 4, 9	Організація роботи в команді. Формати командної роботи	2 тиж. семестру
3,4	<b>Застосування візуального моделювання в процесі розробки ПЗ.</b> Цілі та засоби мови UML. Діаграми взаємодії, класів, станів, діяльності, компонентів, розміщення.	Лекція	3, 5, 6	Розробка вимог до проєкту. Збір вимог до ПЗ та їх аналіз.	4 тиж. семестру
5	<b>Технології створення ПЗ.</b> Загальні вимоги до технологій створення ПЗ. Правила та особливості розробки ПЗ. Проблеми, які виникають при розробці ПЗ.	Лекція	2, 5, 7	Проектування ПЗ. Оцінка тривалості задач проєкту.	6 тиж. семестру
6	<b>Моделі життєвого циклу ПЗ.</b> Каскадна модель. Макетування. Інкрементна модель. Швидка розробка додатків. Спиральна та компонентно-орієнтована моделі. XP-процес.	Лекція	2, 4, 5, 7	Створення проєкту в спеціалізованих сервісах, додавання коду в проєкт, бази даних та розв'язання конфліктів.	7 тиж. семестру
7,8	<b>Класифікація архітектури ПЗ.</b> Аналіз архітектури ПЗ з огляду на область застосування. Монолітна, сервісно-орієнтована, мікросервісна та безсерверна архітектура ПЗ.	Лекція	1, 2, 4, 5	Підсумкове заняття, Контрольний замір знань 1.	8 тиж. семестру

9	<b>Мікросервісна архітектура в контексті веб-розробки.</b> Архітектурний шаблон Модель-Вид-Контролер. Масштабування.	Лекція	4, 5, 8, 9	Реалізація проєкту. Обґрунтування архітектурного стилю. Проведення перегляду коду командою.	10 тиж. семестру
10	<b>Мікросервісна архітектура та контейнеризація.</b> Мікросервісні патерни. Docker-контейнери. Restful web-сервіси. Безпека сервісів.	Лекція	4, 5, 8, 9	Реалізація проєкту. Дослідження використання мікросервісів та Restful.	12 тиж. семестру
11, 12	<b>Проектування мікросервісної архітектури.</b> Оркестрування та відповідні сучасні засоби. Транзакції. Saga патерн.	Лекція	2, 4, 5	Реалізація проєкту. Контейнеризація.	13 тиж. семестру
13	<b>Просторова архітектура.</b> Топологія та модулі обробки. Помпа та колізії даних. Хмарні та локальні реалізації. Характеристики. Віртуалізоване проміжне програмне забезпечення.	Лекція	1, 2, 4	Тестування компонентів проєкту та оркестрування процесів.	14 тиж. семестру
14	<b>Безсерверна архітектура.</b> Хмарні нативні застосунки. Продуктивність. Кешування. AWS та безсерверна архітектура.	Лекція	2, 5, 6, 9	Інтеграція та тестування розробки проєкту.	15 тиж. семестру
15	<b>Вибір архітектурного стилю.</b> Критерії щодо монолітної та розподіленої архітектур. Побудова діаграми рішення.	Лекція	2, 4, 5	Підготовка та створення презентації проєкту	15 тиж. семестру
16	<b>Аналіз якості та оцінка програмного дизайну.</b> Атрибути якості. Методи аналізу архітектури. Комплексний підхід до оцінки архітектури. Метод аналізу вартості та ефективності.	Лекція	2, 4, 5	Підсумкове заняття, Контрольний замір знань 2.	16 тиж. семестру