

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 № *1/24* від *28.08.* 2023 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»

Назва дисципліни	Архітектура програмного забезпечення
Адреса викладання дисципліни	м. Львів, вул. Драгоманова, 50
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіоелектронних і комп'ютерних систем
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 121 Інженерія програмного забезпечення
Викладачі дисципліни	Оленич Ігор Богданович, доктор фіз.-мат. наук, професор Рабош Роман Васильович, кандидат фіз.-мат. наук, асистент
Контактна інформація викладачів	igor olenych@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/olenych-i-broman.rabosh@lnu.edu.ua
Консультації з питань навчання по дисципліні відбуваються	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.
Сторінка дисципліни	https://e-learning.lnu.edu.ua/course/view.php?id=5512
Інформація про дисципліну	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).
Коротка анотація дисципліни	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.
Мета та цілі дисципліни	<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.
Література для вивчення дисципліни	Основна література: 1. Martin R.C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Pearson, 2017. – 432 p. 2. Richards M., N. Ford Fundamentals of Software Architecture. An Engineering Approach. – O'Reilly Media, 2018. – 422 p. 3. Brown S. Visualise, document and explore your software architecture. Software Architecture for Developers. – Leanpub, 2019. – 197 p. 4. Gough J., Bryant D., Auburn M. Mastering API Architecture: Design, Operate, and Evolve API-Based Systems (1st edition). – O'Reilly Media, 2019. – 286 p. 5. Eyskens S. Software Architecture for Busy Developers / S. Eyskens. – Birmingham: Packt Publishing, 2019. – 174 p. Додаткова література: 6. Bewis T. C# Design Pattern Essentials. – NY: Ability First Limited, 2012. – 264 p.

	<p>7. Maciaszek L. Requirements analysis and systems design. – Ontario: Pearson Education Canada, 2007. – 656 p.</p> <p>8. Clements P., Bachmann F., Bass L. Documenting Software Architectures: Views and Beyond (2nd Edition). – Boston: Addison-Wesley Professional, 2010. – 592 p.</p> <p>9. Goma H. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures. – NY: Cambridge University Press, 2011. – 578 p.</p>
<p>Обсяг курсу</p>	<p>Загальний обсяг: 135 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 71 год.</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> - <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. - <i>Вміти</i> виконувати аналіз вимог до програмного забезпечення, що розробляється; оцінювати трудомісткість і вибирати 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. <p>У результаті вивчення курсу «Архітектура програмного забезпечення» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. 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>ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу</p> <p>ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>ФК22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові</p>

	<p>документи в галузі інженерії програмного забезпечення.</p> <p>ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПРН09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПРН11. Вибирати вихідні дані для проектування, куруючись формальними методами опису вимог та моделювання.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.</p> <p>ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p> <p>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.</p>
Ключові слова	Бізнес-логіка, структура програмного забезпечення, дизайн рівнів програмного забезпечення, патерни, CASE засоби.
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ.
Підсумковий контроль, форма	Екзамен у формі захисту командних проектів у кінці семестру.
Пререквізити	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».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація), дедуктивні методи на основі узагальнень, евристичні методи (проблемна лекція), інтерактивні методи (дискусія).
Необхідне обладнання	<p>Для проведення лекційних занять:</p> <ul style="list-style-type: none"> ● монітор TFT 23"; ● системний блок (процесор мінімум 1,8 ГГц, 8 ГБ оперативної пам'яті, 20 ГБ вільного місця на диску); ● мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); ● комутатор мережевий для доступу до мережі Internet. <p>Для проведення лабораторних занять:</p> <ul style="list-style-type: none"> ● комп'ютерна лабораторія з 12-14 робочими місцями; ● монітори TFT 23"; ● системні блоки (процесор мінімум 1,8 ГГц, 8 ГБ оперативної пам'яті, 20 ГБ вільного місця на диску); ● мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); ● комутатор мережевий для доступу до мережі Internet.

	<p>Необхідне програмне забезпечення:</p> <ul style="list-style-type: none"> ● ОС Windows 8 (або вище); ● середовище розробки Visual Studio Express 2019; ● компілятори мов програмування C++/C#; ● Microsoft Teams
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<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"> • лабораторні роботи: 25% семестрової оцінки; максимальна кількість балів 25. • контрольні заміри (2 модулі): 25% семестрової оцінки; максимальна кількість балів 25. • екзамен: 50% семестрової оцінки; максимальна кількість балів 50. <p>Загалом упродовж семестру 100 балів.</p> <hr/> <p>Контрольні заміри проводяться у формі тестових завдань.</p> <p>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 незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>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.</p> <p>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.</p> <p>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 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>Оцінювання лабораторних робіт (5 лабораторних робіт, максимальна кількість балів: 25) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії та захисту звіту по виконаній лабораторній роботі (0-5 балів за одну роботу).</p> <p>Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням:</p> <p>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;</p>

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 модулів (2 тестових модулів, 12.5 балів за кожний) — за результатами написаних студентом тестів.

Бали оцінювання змістових модулів нараховуються за наступним співвідношенням:

10-12.5 – розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, містить аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду повністю робочі та відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;

8-9 – відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведені приклади коду частково робочі, проте в загальному відповідають темі;

6-7 – відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);

4-5 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;

0-3 – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.

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 ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.

Питання до контрольних робіт	Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт розміщені на веб-сторінці курсу.
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	Основні відомості про архітектуру ПЗ. Особливості розробки великих програмних систем. Архітектура і дизайн. Мета архітектури. Функціональність та структура ПЗ. Роль архітектури та архітектора.	Лекція	1, 4, 5, 8	Вступне заняття. Формування команд, обрання теми проєкту.	1 тиж. семестру
2	Парадигми програмування. Структурне програмування. Об'єктно-орієнтоване програмування. Інкапсуляція, успадкування, поліморфізм. Функційне програмування.	Лекція	1, 4, 5	Організація роботи в команді. Формати командної роботи	2 тиж. семестру
3,4	Застосування візуального моделювання в процесі розробки ПЗ. Цілі та засоби мови UML. Діаграми взаємодії, класів, станів, діяльності, компонентів, розміщення.	Лекція	6, 7, 9	Розробка вимог до проєкту. Збір вимог до ПЗ та їх аналіз.	4 тиж. семестру
5	Технології створення ПЗ. Загальні вимоги до технологій створення ПЗ. Правила та особливості розробки ПЗ. Проблеми, які виникають при розробці ПЗ.	Лекція	2, 6, 8	Проектування ПЗ. Оцінка тривалості задач проєкту.	6 тиж. семестру
6	Моделі життєвого циклу ПЗ. Каскадна модель. Макетування. Інкрементна модель. Швидка розробка додатків. Спіральна та компонентно-орієнтована моделі. XP-процес.	Лекція	2, 4, 6, 8	Створення проєкту в спеціалізованих сервісах, додавання коду в проєкт, бази даних та розв'язання конфліктів.	7 тиж. семестру
7,8	Класифікація архітектури ПЗ. Аналіз архітектури ПЗ з огляду на область застосування. Монолітна, сервісно-орієнтована, мікросервісна та безсерверна архітектура ПЗ.	Лекція	1, 4, 6, 8	Підсумкове заняття, Контрольний замір знань 1.	8 тиж. семестру
9	Мікросервісна архітектура в контексті веб-розробки. Архітектурний шаблон Модель-Вид-Контролер. Масштабування.	Лекція	3, 4, 5, 6	Реалізація проєкту. Обґрунтування архітектурного стилю. Проведення перегляду коду командою.	10 тиж. семестру
10	Мікросервісна архітектура та контейнеризація. Мікросервісні патерни. Docker-контейнери. Restful web-сервіси. Безпека сервісів.	Лекція	3, 4, 5, 6	Реалізація проєкту. Дослідження використання мікросервісів та Restful.	12 тиж. семестру
11, 12	Проектування мікросервісної архітектури. Оркестрування та	Лекція	4, 6, 8	Реалізація проєкту. Контейнеризація.	13 тиж. семестру

	відповідні сучасні засоби. Транзакції. Saga патерн.				
13	Просторова архітектура. Топологія та модулі обробки. Помпа та колізії даних. Хмарні та локальні реалізації. Характеристики. Віртуалізоване проміжне програмне забезпечення.	Лекція	1, 4, 8	Тестування компонентів проєкту та оркестрування процесів.	14 тиж. семестру
14	Безсерверна архітектура. Хмарні нативні застосунки. Продуктивність. Кешування. AWS та безсерверна архітектура.	Лекція	5, 6, 7, 8	Інтеграція та тестування розробки проєкту.	15 тиж. семестру
15	Вибір архітектурного стилю. Критерії щодо монолітної та розподіленої архітектур. Побудова діаграми рішення.	Лекція	4, 6, 8	Підготовка та створення презентації проєкту	15 тиж. семестру
16	Аналіз якості та оцінка програмного дизайну. Атрибути якості. Методи аналізу архітектури. Комплексний підхід до оцінки архітектури. Метод аналізу вартості та ефективності.	Лекція	4, 6, 8	Підсумкове заняття, Контрольний замір знань 2.	16 тиж. семестру