



АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА КОМПОНЕНТНО-ОРІЄНТОВАНЕ ПРОЄКТУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 рік підготовки, 2 семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., комп'ютерний практикум: 18 год., самостійна робота: 96 год.</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року (rozklad.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., асистент, Погорелов В.В., volodymyr.pogorelov@gmail.com Комп'ютерний практикум: к.т.н., асистент, Погорелов В.В., volodymyr.pogorelov@gmail.com</i>
Розміщення курсу	<i>Google classroom. Доступ надається зареєстрованим студентам.</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проєктування» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної із проєктуванням програмного забезпечення.

***Метою** вивчення дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проєктування» є вивчення сучасних технологій, методів та засобів проєктування складних програмних систем.*

***Предметом** дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проєктування» є патерни, шаблони та моделі, що використовуються для розроблення складних програмних систем.*

*Вивчення дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проєктування» підсилює у здобувачів освіти **фахові компетентності (ФК)**, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної з проєктуванням архітектури програмних систем:*

ФК03 Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.

ФК05 Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.

ФК09 Здатність забезпечувати якість програмного забезпечення.

ФК10 Здатність проектувати складні мультимедійні та інформаційно-пошукові системи.

ФК11 Здатність проектувати та конструювати, впроваджувати та підтримувати веборієнтовані програмні системи для реалізації нових методів пошуку інформації.

ФК13 Здатність впроваджувати та підтримувати інформаційні системи.

Вивчення дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проектування» сприяє формуванню у здобувачів освіти **програмних результатів навчання (ПРН)** за освітньою програмою:

ПРН03 Будувати і досліджувати моделі інформаційних процесів у прикладній області.

ПРН04 Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення.

ПРН05 Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

ПРН06 Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

ПРН08 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.

ПРН09 Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

ПРН10 Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.

ПРН15 Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

ПРН17 Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

ПРН18 Знати теоретичні засади, що лежать в основі методів досліджень інформаційних систем та програмного забезпечення, методології проведення досліджень та обчислювальних експериментів.

ПРН20 Знати принципи побудови програмних інформаційно-пошукових систем.

ПРН23 Знати та вміти застосовувати на практиці спеціалізовані шаблони проектування інформаційно-пошукових систем.

ПРН24 Вміти проектувати та розробляти мультиагентні інформаційно-пошукові системи.

ПРН25 Вміти проектувати та розробляти розподілені та централізовані інформаційно-пошукові системи.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Успішному вивченню дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проектування» передують вивчення дисциплін «Математичне забезпечення мультимедійних та інформаційно-пошукових систем», «Алгоритмічне забезпечення мультимедійних та інформаційно-пошукових систем», «Алгоритми та структури даних», «Програмування» та «Програмне забезпечення інформаційно-пошукових систем» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Архітектура програмного забезпечення та компонентно-орієнтоване проектування» теоретичні знання та практичні уміння забезпечують успішне

виконання курсових проєктів та магістерських дисертацій за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Архітектура програмного забезпечення та компонентно-орієнтоване проєктування» передбачає вивчення таких тем:

Тема 1. Вступ в архітектуру програмного забезпечення

Тема 2. Проєктування програмних систем

Тема 3. Шаблони архітектур

Тема 4. Патерни GOF

Модульна контрольна робота

Екзамен

4. Навчальні матеріали та ресурси

Базова література:

1. Чиста архітектура: мистецтво розробки програмного забезпечення / Р. Мартін. – Фабула, 2019. – 416 с.
2. Head First. Патерни проєктування/ Е. Фрімен, Е. Робсон – Фабула, 2020. – 672 с.
3. Проєктування інформаційних систем / В.С. Пономаренко, О.І. Пушкар, І.В. Журавльова, С.В. Мінухін. – К.: Видавничий центр «Академія», 2002. – 486 с.
4. Вступ до інженерії програмного забезпечення: навч. посібник / Є.В. Левус, Н.Б. Мельник – Львів: Видавництво Львівської політехніки, 2017. – 280 с.

Додаткова література:

5. Лимаренко Ю. О. Моделювання та аналіз програмного забезпечення: навч.-метод. посібник / Ю. О. Лимаренко – Запоріжжя : ЗДІА, 2014. – 88 с
6. Python, режим доступу: <https://docs.python.org/3/>
7. Патерни проєктування, режим доступу: <https://refactoring.guru/uk/design-patterns>
8. Laplante, Phillip. What Every Engineer Should Know about Software Engineering. Boca Raton: CRC, 2007.
9. Мельник А. Архітектура комп'ютера: підручник. – В-во "Львівська політехніка", Львів, 2009. - 469 с.
10. Тарарака В. Архітектура комп'ютерних систем. Навчальний посібник.— Житомир : ЖДТУ, 2018. 383 с.
11. Табунщик Г. В. Проєктування та моделювання програмного забезпечення сучасних інформаційних систем: навчальний посібник/Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.
12. Null, Linda. Fundamentals of Computer Organization and Architecture (5th Edition). Burlington, Massachusetts: The Teachings of Jones and Bartlett. 2019. - p. 280.
13. Петрик М.Р. Моделювання програмного забезпечення : науковометодичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Видво ТНТУ імені Івана Пулюя, 2015. – 200 с
14. Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: навчальний посібник / І.М. Дудзяний. – Львів, Видавничий центр ЛНУ ім. І. Франка. 2007. – 108 с.,
15. Грицюк Ю. І. Аналіз вимог до програмного забезпечення: навчальний посібник. / Ю. І. Грицюк – Львів : Видавництво Львівської політехніки, 2018. – 456 с.

Використати для опанування практичних умінь дисципліни. Матеріали знаходяться у вільному доступі в Інтернеті.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Вступ в архітектуру програмного забезпечення</i>		
1	<i>Лекція 1. Вступ до дисципліни; створення складних програмних систем; термінологія; проблематика</i>	<i>Причина виникнення науки та дисципліни «архітектура програмного забезпечення». Основна термінологія. Принципологія та концепції створення складних програмних систем Завдання на СРС: п.6 №1.</i>
2	<i>Лекція 2. Архітектура програмних систем; типи архітектур</i>	<i>Поняття архітектури програмних систем. Основні типи архітектур програмного забезпечення. Завдання на СРС: п.6 №2.</i>
3	<i>Лекція 3. Модульність та декомпозиція в проектуванні програмних систем</i>	<i>Поняття модуля програмної системи. Модульність в проектуванні програмного забезпечення. Декомпозиція, правила декомпозиції при побудові складних програмних систем Завдання на СРС: п.6 №3.</i>
4	<i>Лекція 4. Життєвий цикл програмного продукту; моделі; стадії; поняття</i>	<i>Поняття та вимоги до життєвого циклу програмного забезпечення. Моделі життєвого циклу. Основні стадії та особливості життєвого циклу програмного забезпечення для моделей Завдання на СРС: п.6 №4.</i>
<i>Тема 2. Проектування програмних систем</i>		
5	<i>Лекція 5. Постановка вимог та цілей до архітектури програмних систем; аналіз вимог та розробка зовнішнього представлення проєкту</i>	<i>Підходи програмного інженера на етапі постановки вимог до розробки програмних систем. Комунікативна складова програмного інженера на етапі постановки вимог. Зовнішнє представлення та аналіз макету проєкту Завдання на СРС: п.6 №5.</i>
6	<i>Лекція 6. Проектування архітектури програмних систем, методи та методологія</i>	<i>Основні підходи до проектування складних програмних систем. Стили та підходи до проектування програмного забезпечення, розроблення та прийняття проєктних рішень програмного інженера. Завдання на СРС: П.6 №6.</i>
7	<i>Лекція 7. Аналіз архітектури програмного проєкту</i>	<i>Огляд та застосування методу аналізу компромісних архітектурних рішень АТАМ для високонавантажених програмних систем</i>

		<i>Завдання на СРС: п.6 №7.</i>
8	<i>Комп'ютерний практикум 1. Побудова архітектури програмної системи</i>	<i>Завдання: Проаналізувати вимоги до програмної системи та спроєктувати відповідну їм архітектуру</i> <i>Завдання на СРС: п.6 №8.</i>
<i>Тема 3. Шаблони архітектур</i>		
9	<i>Лекція 8. Атрибути якості програмного продукту</i>	<i>Визначення атрибутів якості програмних систем на основі аналізу нефункціональних вимог до розроблення програмного забезпечення.</i> <i>Завдання на СРС: п.6 №9.</i>
10	<i>Лекція 9. Архітектурні шаблони; вибір архітектурних шаблонів при проєктуванні складних програмних систем</i>	<i>Методологія вибору архітектурних патернів та подальше їх застосування в складних програмних системах. Патерни: клієнт-сервер, майстер-робітники, сервіс-орієнтована архітектура, гнучка архітектура, канали та фільтри.</i> <i>Завдання на СРС: п.6 №10.</i>
11	<i>Лекція 10. Багатошаровий принцип побудови архітектури програмного забезпечення</i>	<i>Представлення архітектури системи у вигляді сукупності архітектурних шарів. Класична багатошарова модель, модель орен-layer.</i> <i>Завдання на СРС: п.6 №11.</i>
12	<i>Лекція 11. Патерни проєктування нижнього рівня; класифікація; принципи застосування</i>	<i>Огляд основних низькорівневих патернів проєктування (design patterns). Наведення їх класифікації за трьома типами. Прийоми та правила застосування патернів низькорівневого проєктування кожного з типів.</i> <i>Завдання на СРС: п.6 №12.</i>
13	<i>Комп'ютерний практикум 2. Проєктування архітектурних шарів програмної системи</i>	<i>Завдання: Створити програмну систему з багатошаровою архітектурою за класичною багатошаровою моделлю.</i> <i>Завдання на СРС: п.6 №13.</i>
<i>Тема 4. Патерни GOF</i>		
14	<i>Лекція 13. Патерни, що породжують</i>	<i>Огляд та правила застосування патернів, що призначені для створення нових об'єктів в програмній системі. Builder, Abstract factory, Singleton.</i> <i>Завдання на СРС: п.6 №14.</i>
15	<i>Лекція 14. Структурні патерни</i>	<i>Огляд та правила застосування патернів, призначених для компонування програмної системи на основі об'єктів та класів. Розглядаються наступні структурні патерни: Adapter, Facade, Bridge.</i>

		<i>Завдання на СРС: п.6 №15.</i>
16	<i>Лекція 15. Патерни поведінки</i>	<i>Огляд та правила застосування патернів, призначених для регламентації поведінки об'єктів в програмній системі. Розглядаються наступні патерни поведінки: Iterator, Template method, Strategy. Завдання на СРС: п.6 №16.</i>
17	<i>Лекція 16. Оцінка якості архітектури програмного продукту</i>	<i>Оцінка відповідності архітектури програмної системи до нефункціональних вимог та визначених атрибутів якості. Завдання на СРС: п.6 №17.</i>
18	<i>Комп'ютерний практикум 3. Застосування низькорівневих шаблонів проектування</i>	<i>Завдання: побудувати систему з використанням патернів GOF. Завдання на СРС: п.6 №18.</i>
<i>Модульна контрольна робота</i>		

6. Самостійна робота студента/аспіранта

Дисципліна «Архітектура програмного забезпечення та компонентно-орієнтоване проектування» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

<i>№ з/п</i>	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>Кількість годин</i>	<i>Література</i>
1	<i>Підготовка до лекції 1</i>	2	1; 2; 3; 8
2	<i>Підготовка до лекції 2</i>	2	1; 4, 7
3	<i>Підготовка до лекції 3</i>	2	1; 3; 5; 9; 10
4	<i>Підготовка до лекції 4</i>	2	11-13
5	<i>Підготовка до лекції 5</i>	2	3; 6; 10
6	<i>Підготовка до лекції 6</i>	2	3; 8; 10
7	<i>Підготовка до лекції 7</i>	2	11-13
8	<i>Підготовка до комп'ютерного практикуму 1</i>	7	2; 5
9	<i>Підготовка до лекції 8</i>	2	2; 9
10	<i>Підготовка до лекції 9</i>	2	11-13
11	<i>Підготовка до лекції 10</i>	2	2; 6
12	<i>Підготовка до лекції 11</i>	2	1; 2; 3
13	<i>Підготовка до комп'ютерного практикуму 2</i>	7	11-13
14	<i>Підготовка до лекції 12</i>	2	4; 5
15	<i>Підготовка до лекції 13</i>	2	4; 5
16	<i>Підготовка до лекції 14</i>	2	9-10

17	Підготовка до лекції 15	2	4, 12
18	Підготовка до лекції 16	2	5; 15
19	Підготовка до комп'ютерного практикуму 3	7	3, 11-13
20	Підготовка до модульної контрольної роботи	13	1-15
21	Підготовка до екзамену	30	1-15

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим.

Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби консультації/захисту робіт комп'ютерного практикуму.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту робіт комп'ютерного практикуму: роботи повинні бути зроблені відповідно до поставлених задач та згідно з варіантом.

Правила призначення заохочувальних та штрафних балів є наступними. Заохочувальні бали нараховуються за:

- точні та повні відповіді в опитуваннях за матеріалами лекцій (максимальна кількість балів за блиц-опитування - 3 бали).

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 3 комп'ютерних практикуми. Максимальна кількість балів за кожний комп'ютерний практикум: 10 балів.

Бали нараховуються за:

- якість виконання комп'ютерного практикуму: 0-6 бали;*
- відповідь під час захисту комп'ютерного практикуму: 0-2 бали;*
- своєчасне представлення роботи до захисту: 0-2 бали.*

Критерії оцінювання якості виконання:

- 6 балів – робота виконана якісно, в повному обсязі;*
- 4-5 бали – робота виконана якісно, в повному обсязі, але має недоліки;*
- 2-3 бали – робота виконана в повному обсязі, але містить незначні помилки;*
- 0-1 бал – робота виконана не в повному обсязі.*

Критерії оцінювання відповіді:

- 2 бали – відповідь повна, добре аргументована;*
- 1 бал – у відповіді є суттєві помилки;*
- 0 балів – немає відповіді або відповідь невірна.*

Критерії оцінювання своєчасності представлення роботи до захисту:

- 2 бали – робота представлена до захисту не пізніше вказаного терміну;*
- 0 балів – робота представлена до захисту пізніше вказаного терміну.*

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:

10 балів × 3 комп. практ. = 30 балів.

Завдання на модульну контрольну роботу складається з 2 теоретичних та 2 практичних запитань. Відповідь на кожне запитання оцінюється 5 балами.

Критерії оцінювання кожного запитання контрольної роботи:

4-5 балів – відповідь вірна, повна, добре аргументована;
2-3 балів – в цілому відповідь вірна, але має недоліки;
1 бал – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

5 балів × 4 запитання = 20 балів.

Рейтингова шкала з дисципліни дорівнює:

$R = R_C = R_{\text{ком.практ}} + R_{\text{МКР}} = 30 \text{ балів} + 20 \text{ балів} = 50 \text{ балів.}$

$R = R_C + R_{\text{екзамен}} = 50 \text{ балів} + 50 \text{ балів} = 100 \text{ балів.}$

Екзаменаційна складова рейтингової шкали $R_{\text{екзамен}} = 50$ балів.

Завдання на екзамен складається з 5 питань – 3 теоретичних та 2 практичних завдань.

Відповідь на кожне теоретичне та практичне запитання оцінюється 10 балами.

Критерії оцінювання теоретичного/практичного питання екзаменаційної роботи:

9-10 балів – відповідь вірна, повна, добре аргументована;

7-8 балів – відповідь вірна, але погано аргументована ;

5-6 балів – відповідь вірна, але неповна та погано аргументована;

3-4 бали – у відповіді є незначні помилки;

1-2 бали – у відповіді є суттєві помилки;

0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за екзаменаційну роботу:

10 балів × 3 теоретичні запитання + 10 балів × 2 практичні запитання = 50 балів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силябусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 5 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 10 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: екзамен

Умови допуску до семестрового контролю:

При семестровому рейтингу (R_C) не менше 30 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент допуск до екзамену. Після складання екзамену виставляється оцінка відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою).

Необхідною умовою допуску до екзамену є виконання і захист комп'ютерного практикуму.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., асистент, Погорелов В.В.

Ухвалено кафедрою ПЗКС (протокол №8 від 25.01.2023)

Погоджено Методичною комісією факультету прикладної математики (протокол №6 від 27.01.2023)

Додаток 1. Перелік питань, які виносяться на семестровий контроль

- 1. Поняття «архітектура програмного забезпечення», «програмна система», «проектування програмної системи», «програмна інженерія».*
- 2. Роль архітектора програмної архітектури. Комунікативні зв'язки архітектора програмного забезпечення.*
- 3. Фактори, що впливають на вибір архітектури програмного забезпечення.*
- 4. Основні типи програмних архітектур. Їх класифікація за типами.*
- 5. Поняття «Модульні види архітектури програмного забезпечення».*
- 6. Модульна розробка програмних систем мовою Python. Створення модулів мовою Python.*
- 7. Поняття життєвого циклу програмного забезпечення. Основні етапи життєвого циклу.*
- 8. Проектування архітектури програмних систем. Архітектурний стиль.*
- 9. Методи оцінки складності ієрархічних структур.*
- 10. Методи аналізу програмної архітектури.*
- 11. Атрибути якості архітектури складних програмних систем.*
- 12. Архітектурні рівні програмної системи. Відкриті та закриті багаторівневі архітектури.*
- 13. Архітектурні патерни програмного забезпечення.*
- 14. Класифікація патернів проектування нижнього рівня.*
- 15. Значення архітектури програмного забезпечення при проектуванні програмних систем.*
- 16. Породжуючі патерни.*
- 17. Структурні патерни.*
- 18. Поведінкові патерни.*
- 19. Архітектурний шаблон «Одинак».*
- 20. Архітектурний шаблон «Стратегія».*
- 21. Декомпозиція та правила декомпозиції програмних модулів.*