



Силабус освітнього компонента Програма навчальної дисципліни



Архітектура та проектування програмного забезпечення

Шифр та назва спеціальності

F6 – Інформаційні системи та технології

Спеціалізація

Освітня програма

Програмне забезпечення інформаційних систем

Рівень освіти

Перший (бакалаврський)

Семестр

5, 6

Інститут

ННІ Комп'ютерних наук та інформаційних технологій

Кафедра

Інформаційні системи та технології (329)

Тип дисципліни

Спеціальна (фахова), Обов'язкова

Форма навчання

Денна,

Мова викладання

Українська

Викладачі, розробники



Пашнев Андрій Анатолійович

Andrey.Pashnev@khpi.edu.ua

Кандидат технічних наук, старший науковий співробітник, доцент кафедри ICT НТУ «ХПІ»

Підготував та опублікував понад 150 наукових та навчально-методичних праць (Google Scholar:

<https://scholar.google.com/citations?user=KcBe4YwAAAAJ&hl=uk>;

ORCID: <https://orcid.org/0000-0002-9150-6108>).

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Під час вивчення навчальної дисципліни студенти будуть мати можливість ознайомитись з основами проектування програмного забезпечення, принципами та методами структурного аналізу об'єктів проектування, вимогами до програмного забезпечення, архітектурними стилями побудови програмних систем, основними архітектурними патернами та принципами побудови програмних систем, патернами та принципами проектування програмного забезпечення, корпоративними патернами, порядком оцінювання та документування архітектури програмного забезпечення, стадіями і моделями життєвого циклу програмних систем, основними методологіями розробки програмного забезпечення, основними програмними інструментами проектування, документування та керування архітектурою програмного забезпечення.

Мета та цілі дисципліни

Надання студентам базових знань щодо основ проектування програмного забезпечення, структурного аналізу об'єктів проектування та формування вимог до програмного забезпечення,

сутності та можливостей застосування архітектурних стилів програмних систем, ознайомлення студентів з принципами побудови архітектури програмного забезпечення та її проектування, сутністю та можливостями застосування патернів проектування програмних систем та корпоративних патернів, моделей життєвого циклу та методологій розробки програмних систем, програмними інструментами проектування, документування та керування архітектурою програмного забезпечення та порядком її оцінювання.

Формат занять

Лекції, лабораторні роботи, курсовий проект (2 семестр), самостійна робота, консультації. Підсумковий контроль – іспит (1 семестр), іспит (2 семестр).

Компетентності

КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 7. Здатність розробляти та управляти проектами.

КС 1. Здатність аналізувати об'єкт проектування або функціонування та його предметну область.

КС 2. Здатність застосовувати стандарти в області інформаційних систем та технологій при розробці функціональних профілів, побудові та інтеграції систем, продуктів, сервісів і елементів інфраструктури організації.

КС 3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 7. Здатність застосовувати інформаційні технології у ході створення, впровадження та експлуатації системи менеджменту якості та оцінювати витрати на її розроблення та забезпечення.

КС 10. Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводження інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.

Результати навчання

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПР 4. Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.

ПР 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

ПР 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

ПР 7. Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

ПР 8. Застосовувати правила оформлення проектних матеріалів інформаційних систем та технологій, знати склад та послідовність виконання проектних робіт з урахуванням вимог відповідних нормативно-правових документів для запровадження у професійній діяльності.

Обсяг дисципліни

Загальний обсяг дисципліни 240 год. (8 кредитів ECTS): лекції – 48 год., лабораторні роботи – 64 год., самостійна робота – 128 год.

Передумови вивчення дисципліни (пререквізити)

Об'єктно-орієнтоване програмування. Ознайомча практика. Бази даних. Основи веб-розробки. Аналітика бізнес систем.

Особливості дисципліни, методи та технології навчання

Інтерактивні лекції з презентаціями, дискусії, лабораторні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання..

Програма навчальної дисципліни

Навчальні заняття

Лекції

Семестр 1

Теми лекцій	Кількість годин
Тема 1. Сутність архітектури та проектування програмного забезпечення Класифікація програмних систем. Призначення архітектури програмної системи та рівні її абстракції. Сутність проектування програмної системи.	2
Тема 2. Структурний аналіз об'єктів проектування Сутність та основні принципи структурного аналізу об'єктів проектування. Метод моделювання потоків даних DFD.	2
Тема 3. Вимоги до програмного забезпечення інформаційних систем Класифікація вимог до програмного забезпечення. Техніки збирання вимог та їх аналізу, типові проблеми. Учасники процесу роботи з вимогами. Специфікація програмних вимог. Перевірка вимог.	2
Тема 4 Архітектурні стилі розробки програмного забезпечення Архітектурні стилі в розробці програмного забезпечення. Розподілена архітектура: архітектура клієнт-сервер, архітектура однорангового зв'язку (P2P). Структурна архітектура: компонентна архітектура, монолітна архітектура, багаторівнева архітектура. Архітектурні шаблони обміну повідомленнями: шаблон публікації-підписки, архітектура керована подіями.	4
Тема 5 Архітектурні патерни Сервісно-орієнтована архітектура. Патерн CQRS (Command Query Responsibility Segregation). Доменно-керований патерн. Патерн MVC (Model-View-Controller). Мікросервісна архітектура. Патерн Blackboard. Патерн мікроядра. Безсерверна архітектура. Патерн черги повідомлень. Патерн джерела подій.	6
Загальна кількість годин	16

Семестр 2

Теми лекцій	Кількість годин
Тема 1. Архітектурні принципи побудови програмних систем Принципи компонентів. Політика проти деталей. Зчеплення та згуртованість.	
Тема 2. Принципи проектування програмного забезпечення Композиція понад успадкуванням. Інкапсуляція того, що змінюється. Програмування проти абстракцій. Голлівудський принцип. Принципи SOLID. Принцип DRY (Don't Repeat Yourself). Принцип YAGNI (You Ain't Gonna Need It).	
Тема 3. Патерни проектування програмних систем Класифікація патернів проектування. Патерни проектування GoF (Gang of Four): патерни створення, структурні патерни, патерни поведінки. Патерни проектування GRASP (General Responsibility Assignment Software Patterns). Патерни проектування POSA (Patterns of Scalable and Adaptable Software Architecture).	2
Тема 4. Корпоративні патерни Патерн об'єкта передачі даних. Патерн ідентифікаційної карти. Патерни випадків використання. Патерн репозиторій. Патерн відображення даних. Патерн сценарію транзакції. Патерн об'єктів значення. Патерн моделі домену. Патерн сутності. Області застосування ORM (Object-Relational Mapping).	6
Тема 5. Оцінка архітектури програмного забезпечення Мета аналізу архітектури. Метрики програмного забезпечення для аналізу архітектури. Інструменти та методи аналізу архітектури. Архітектурна ерозія. Забезпечення дотримання архітектурних основ та моделей під час розробки. Майбутнє цифрових архітектур.	6
Тема 6. Документація архітектури програмного забезпечення Цілі документування архітектури програмного забезпечення. Полотно архітектури програмного забезпечення. Структуризація документації з архітектури програмного забезпечення. Шаблон arc42. Альтернативні підходи до документації та структурування. Візуалізація та керування документацією з архітектури програмного забезпечення. Модель C4. Вимоги до документації. Діаграми як код.	6
Тема 7. Життєвий цикл програмної системи Поняття життєвого циклу програмної системи. Основні стадії життєвого циклу та їх взаємозв'язок. Основні моделі життєвого циклу програмної системи. Їх характеристика, переваги та ризики застосування.	6
Тема 8. Основні методології розробки програмного забезпечення Класифікація методологій розробки програмного забезпечення. Порівняльна характеристика адаптивних та прогнозованих методологій розробки програмного забезпечення. Методологія RUP (Rational Unified Process). Методологія XP (Extreme Programming). Методологія SCRUM. Методологія KANBAN.	2
Тема 9. Програмні інструменти проектування, документування та керування архітектурою програмного забезпечення Розподілена система контролю версій. GitHub. Slack. Trello. Atlassian Tools.	4
Загальна кількість годин	32

Практичні заняття

Практичні заняття в рамках дисципліни не передбачені.

Лабораторні заняття

Семестр 1

Теми лабораторних занять	Кількість годин	Вагові коефіцієнти a
Тема 1. Аналіз об'єкта проектування та його предметної області	4	0
Тема 2. Моделювання потоків даних за допомогою методу DFD	6	0
Тема 3. Розробка специфікації вимог до програмного забезпечення	6	0
Тема 4 Розробка системної архітектури програмного Web-застосунку з використанням архітектурного стилю клієнт-сервер	8	0
Тема 5 Розробка системної архітектури програмного Web-застосунку з використанням архітектурного патерна MVC	8	0
Загальна кількість годин	32	$\sum_{i=1}^n a_i$

Семестр 2

Теми лабораторних занять	Кількість годин	Вагові коефіцієнти a
Тема 1. Аналіз та реалізація архітектурних принципів побудови програмних систем	4	0
Тема 2. Реалізація принципів SOLID	4	0
Тема 3. Вибір та реалізація патернів проектування GoF	4	0
Тема 4. Вибір та реалізація корпоративних патернів	4	0
Тема 5. Оцінка розробленої архітектури програмного Web-застосунку	4	0
Тема 6. Розробка документації архітектури програмного Web-застосунку	4	0
Тема 7. Визначення моделі життєвого циклу програмного Web-застосунку	2	0
Тема 8. Вибір методології розробки програмного Web-застосунку	2	0
Тема 9. Документування та керування архітектурою програмного Web-застосунку	4	0
Загальна кількість годин	32	$\sum_{i=1}^n a_i$

Контрольні роботи

Теми контрольних робіт

Вагові
коефіцієнти b

Тема 1. Архітектурні патерни	0
Тема 2. Патерни проектування програмних систем	0
Загалом	$\sum_{i=1}^m b_i$

Самостійна робота

Опрацювання теоретичного матеріалу

Теми для самостійного вивчення

Кількість годин

Тема 1. Огляд необхідних навичок архітектора програмного забезпечення	4
Тема 2. Аналіз можливостей програмних засобів моделювання бізнес-процесів	4
Тема 3 Огляд прикладів специфікації вимог до програмного забезпечення (SRS)	4
Тема 4. Аналіз прикладів реалізації компонентної, монолітної та багаторівневої архітектур	4
Тема 5 Аналіз прикладів реалізації сервісно-орієнтованої та мікросервісної архітектур	4
Тема 6. Аналіз прикладів реалізації архітектурного принципу політика проти деталей	4
Тема 7. Аналіз прикладів реалізації принципів проектування DRY та YAGNI	4
Тема 8. Аналіз прикладів реалізації патернів проектування GRASP та POSA	4
Тема 9. Аналіз прикладів реалізації корпоративного патерна сутності	4
Тема 10. Огляд прикладів архітектурної ерозії програмних систем	4
Тема 11. Додаткові інструменти для документування архітектури програмного забезпечення	6
Тема 12. Порівняльний аналіз основних моделей життєвого циклу програмної системи	4
Тема 13. Порівняльний аналіз адаптивних методологій розробки програмного забезпечення	4
Тема 14. Порівняльний аналіз можливостей GitLab та GitHub	4
Загальна кількість годин	58

Тематика індивідуальних завдань

Планом передбачено курсовий проект в 2 семестрі.

Теми індивідуального завдання

Тема курсового проекту.

Під час виконання курсового проекту необхідно спроектувати програмний Web-застосунок, який дозволяє розв'язати певну задачу.

Вимоги до реалізації проекту:

- 1) на основі сутностей предметної області створити класи, які їм відповідають;
- 2) класи і методи повинні мати назви, що відображають їх функціональність, і повинні бути рознесені по пакетах;
- 3) інформацію щодо предметної області зберігати у реляційній базі даних;
- 4) архітектура застосунка повинна відповідати шаблону MVC;
- 5) при реалізації бізнес-логіки необхідно використовувати шаблони проектування: Команда, Стратегія, Фабрика, Будівельник, Сінглтон, Фронт-контролер, Спостерігач, Адаптер та ін.; використання шаблонів повинно бути обґрунтованим;
- 6) реалізувати функціональність, наведену в постановці завдання.

Тема курсового проекту: проектування програмного Web-застосунку для обраної предметної області.

Загальна кількість годин

70

Неформальна освіта

Рекомендовані курси, тренінги, стажування

1. Онлайн курс: <https://prometheus.org.ua/prometheus-free/git-for-distributed-software/> - зарахування теми 9 у другому семестрі.

Література, навчальні матеріали та інформаційні ресурси

Основна література

- 1 Піхлер Р. Agile продукт-менеджмент за допомогою Scrum. Фабула, 2019. – 128 с.
- 2 Мартін Роберт. Чистий код. – Фабула, 2019. – 416 с.
- 3 Fred Heath. Managing Software Requirements the Agile Way. – Packt Publishing, 2020. – 214 p.
- 4 Charles Tatum. THE ART AND SCIENCE OF SOFTWARE DEVELOPMENT. – Winthrop Publishers, 2020. – 125 p.
- 5 Gabriel Baptista, Francesco Abbruzzese. Software Architecture with C# 9 and .NET 5. Second Edition. – Packt Publishing, 2020. – 701 p.
- 6 Roger S. Pressman, Bruce R. Maxim. SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH. – McGraw-Hill Education, 2020. – 705 p.
- 7 Umesh Kumar Tiwari, Santosh Kumar. Component-Based Software Engineering. Methods and Metrics. – Taylor & Francis Group, LLC, 2021. – 226 p.
- 8 Boyd L. Summers. Effective Methods for Software Engineering. – Taylor & Francis Group, LLC, 2021. – 183 p.
- 9 Fabio Cicerchia. 10x Software Engineer. Curated contents for software engineers. – Lean Publishing, 2021. – 249 p.

Додаткова література

- 1 Freeman Eric, Robson Elisabeth. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition. – O'Reilly Media, 2020. – 672 p.
- 2 Scott A. Whitmire. Engineer Your Software! – Morgan & Claypool Publishers, 2021. – 145 p.
- 3 Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghan. Software Architecture: The Hard Parts. – O'Reilly Media, 2021. – 462 p.
- 4 Thomas Kilian. Inside Enterprise Architect. – Lean Publishing, 2021. – 127 p.
- 5 Paul Beynon-Davies. BUSINESS INFORMATION SYSTEMS. THIRD EDITION. – Red Globe Press, 2020. – 511 p.
- 6 Capers Jones. Software Development Patterns and Antipatterns. – CRC Press, 2021. – 513 p.
- 7 John D. McDowall, Complex Enterprise Architecture: A New Adaptive Systems Approach. – APRESS, 2019. – 164 p.
- 8 Carola Lilienthal. Sustainable Software Architecture. Analyze and Reduce Technical Debt. – dpunkt.verlag, 2019. – 309 p.

- 9 Daniel Heller. Building a Career in Software: A Comprehensive Guide to Success in the Software Industry. – APRESS, 2020. – 240 p.
- 10 Murat Erder, Pierre Pureur, Eoin Woods. Continuous Architecture in Practice. Software Architecture in the Age of Agility and DevOps. – Pearson Education, 2021. – 353 p.
- 11 Jocelyn O. Padallan. Distributed Database Architecture. – Arcler Press, 2021. – 266 p.
- 12 Tsitoara Mariot. Beginning Git and GitHub. – APRESS, 2020. – 294 p.
- 13 Chris Belanger. Mastering Git. – Razeware LLC, 2019. – 213 p.
- 14 Guide to the Software Engineering Body of Knowledge. Version 3.0.

Система оцінювання

Підсумкова оцінка з освітнього компонента визначається відповідальним лектором за темами, видами занять, тощо у відповідності до силабусу і є інтегральною оцінкою результатів усіх вид навчальної діяльності здобувача вищої освіти. Підсумкова оцінка повинна відображати всі оцінки за складовими навчального процесу з урахуванням їх вагових показників k :

Поточний контроль (практичні, семінарські, лабораторні заняття), k_1	Контрольні роботи (за наявності), k_2	Індивідуальне завдання (за наявності), k_3	Підсумковий контроль (для ОК з іспитом), k_4
0,6	0,2	0,2	0,0

Сума коефіцієнтів повинна складати одиницю: $k_1 + k_2 + k_3 + k_4 = 1$. Підбір вагових коефіцієнтів підсумкової оцінки здійснює розробник курсу.

Розрахунок підсумкової оцінки проводиться за формулою:

$$O = П \cdot k_1 + K \cdot k_2 + I \cdot k_3 + Пк \cdot k_4$$

де: $П$ – середньозважена середня оцінка за поточний контроль

I – оцінка за виконання індивідуального завдання

K – середньозважена оцінка за контрольні роботи

$Пк$ – оцінка за підсумковий контроль

$$П = \frac{П_1 \cdot a_1 + П_2 \cdot a_2 + \dots + П_n \cdot a_n}{\sum_{i=1}^n a_i}$$

де: a_i - ваговий коефіцієнт за кожне практичне (семінарське) або лабораторне заняття.

$$K = \frac{K_1 \cdot b_1 + K_2 \cdot b_2 + \dots + K_m \cdot b_m}{\sum_{i=1}^m b_i}$$

де: b_i - ваговий коефіцієнт за кожну контрольну роботу.

Шкала оцінювання

Поточні оцінки за кожну складову ($П, K, I, \dots$) виставляються за 100-бальною шкалою згідно з [положенням «Про критерії та систему оцінювання знань та вмінь і про рейтинг здобувачів вищої освіти» НТУ «ХПІ»](#).

Підсумкова оцінка виставляється відповідно до розрахованої O з округленням до найближчого цілого числа в більшу сторону.

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Здобувач вищої освіти повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»»: виявляти дисциплінованість, вихованість, доброзичливість, чесність,

відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту.

Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

26.08.2025

Завідувач кафедри

Олена НІКУЛІНА

26.08.2025

Гарант ОП

Ірина ЛЮТЕНКО