



Компоненти програмної інженерії – архітектура програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Заочна</i>
Рік підготовки, семестр	<i>2 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредита (120 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Другий семестр</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>асистент Смілянець Ф.А.</i> , Лабораторні: <i>асистент Смілянець Ф.А.</i>
Розміщення курсу	Google Classroom, https://classroom.google.com/c/NzgwNDYxMDM4OTQ2?cjc=efg7olom

Програма навчальної дисципліни

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

Вивчення дисципліни спрямовано на оволодіння компетентностями з прийняття архітектурних рішень в процесі розробки програмного забезпечення. Увага приділяється документуванню архітектури та архітектурним патернам.

Предмет навчальної дисципліни – прийняття архітектурних рішень в циклі розробки програмного забезпечення.

Метою дисципліни є розуміння студентами процесу побудови архітектури та ролі архітектора.

Програмні результати навчання студента.

Студент після засвоєння навчальної дисципліни повинен **знати**:

- Призначення процесу побудови архітектури
- Місце архітектури в циклі розробки програмного забезпечення

- Основи та мета документування архітектури, засоби документування
- Основні сучасні архітектурні патерни

Студент після засвоєння навчальної дисципліни повинен **вміти**:

- Приймати архітектурні рішення
- Документувати архітектурні рішення
- Комунікувати архітектурні рішення

Набуті знання та навички можна використати при працевлаштуванні.

Відповідність навчальної дисципліни освітньо-професійній програмі (фахові компетентності ФХ та програмні результати навчання ПРН):

1. ФК2 проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
2. ФК3 Здатність розробляти архітектури, модулі та компоненти програмних систем.
3. ПРН 12 Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
4. ПРН 23 Вміти документувати та презентувати результати розробки програмного забезпечення.

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

Вивчення дисципліни спирається на знання, отримані студентами при вивченні дисциплін «Об'єктно-орієнтоване програмування», «Веб-орієнтована розробка програмного забезпечення», мови програмування. Набуті знання та навички можна використати при працевлаштуванні.

3. Зміст навчальної дисципліни

Тема 1	Поняття архітектури, її місце в циклі розробки програмного забезпечення
Тема 2	Документування архітектури
Тема 3	Архітектурні точки зору (architecture view)
Тема 4	Якісні атрибути(нефункціональні вимоги)
Тема 5	Архітектурі тактики та патерни
Тема 6	Архітектурний патерн «Моноліт. COA. Мікросервіси»
Тема 7	Архітектурний патерн «Мікросервіси»
Тема 8	Архітектурний патерн «Мікросервіси» (продовження)
Тема 9	Архітектурний патерн «Event-driven architecture»

Тема 10	Безсерверна архітектура
Тема 11	Архітектурний патерн «Space-based architecture». Кешування
Тема 12	Масштабування
Тема 13	Патерни відмовостійкості та надійності
Тема 14-15	Безпека
Тема 16	Стратегії випуску та розгалуження (release and branching strategies)
Тема 17	Типи збереження даних
Тема 18	Кластерні системи. CAP теорема

4. Навчальні матеріали та ресурси

Основна

1. Software Architecture in Practice (SEI Series in Software Engineering) 3rd Edition. Len Bass, Paul Clements, Rick Kazman. Publisher: Addison-Wesley. 2012. 624 pages.
2. Designing Software Architectures: A Practical Approach. Humberto Cervantes, Rick Kazman, May 2016. Publisher: Addison-Wesley.
3. Documenting Software Architectures. Views and Beyond. 2nd Edition Paul Clements, Felix Bachmannl, Len Bass, David Garlan, James Ivers. Addison Wesley, 2011. 592 pages.
4. Fundamentals of Software Architecture: A Comprehensive Guide to Patterns, Characteristics, and Best Practices 1st Edition. Neal Ford, Mark Richards. O'Reilly Media. 2020. 500 pages

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді у Google Classroom, до якого надано доступ групі студентів та асистентам, які ведуть заняття комп'ютерного практикуму.

Лекції по дисципліні проводяться викладачем із використанням сучасних мультимедійних презентаційних технологій.

В умовах дистанційного навчання усі види занять, у тому числі контрольні заходи, проводяться в Google Meet.

5.1 Тематика лекцій

Тема 1	Поняття архітектури, її місце в циклі розробки програмного забезпечення	1. Definition and purpose 2. Architecture in SDLC. 3. Architecture in preparing sprints in Agile/Scrum. 4. Organizing architecture-related communication.
--------	---	--

Тема 2	Документування архітектури	1. Documenting architecture. 2. Architecture document template. Simplest architecture document example (with system decomposition and API design).
Тема 3	Архітектурні точки зору (architecture view)	1. C4 views 2. 4+1 views 3. Module views, component-and-connector views, allocation views 4. Notations to present software architecture.
Тема 4	Якісні атрибути(нефункціональні вимоги)	1. Quality attributes 2. Quality attribute scenarios 3. Availability 4. Performance 5. Security 6. Modifiability etc
Тема 5	Архітектурі тактики та патерни	1. Tactics 2. Patterns 3. Pattern "Layers" 4. Other patterns
Тема 6	Архітектурний патерн «Моноліт. COA. Мікросервіси»	1. Monolith architecture style 2. SOA architecture style 3. Microservices architecture style 4. Why microservices 5. Decomposing to microservices 6. Challenges of microservices architecture - working with data
Тема 7	Архітектурний патерн «Мікросервіси»	1. Data: migration scripts 2. Data sync between microservices 3. Data aggregation for microservices
Тема 8	Архітектурний патерн «Мікросервіси» (продовження)	1. Data patterns: distributed transactions with Saga pattern 2. Compute patterns 3. Communication patterns (sync REST, async REST, messaging: 1-1, 1-N) 4. Operate: API Gateway, Service Discovery, Externalized configuration
Тема 9	Архітектурний патерн «Event-driven architecture»	1. Event driven architecture concept 2. Event-driven architecture patterns - Event sourcing - Streaming analytics - CQRS - Broker vs Mediator
Тема 10	Безсерверна архітектура	1. What is serverless? Why serverless? Serverless vs microservices 2. Use cases 3. Patterns 4. Considerations 5. Serverless Framework
Тема 11	Архітектурний патерн «Space-based architecture». Кешування	1. Space-based architecture 2. Caching patterns 3. Caching topologies
Тема 12	Масштабування	1. Scaling cube

		2. Vertical vs Horizontal scaling 3. Replication vs sharding. 4. Scaling examples (Redis, PostgreSQL, MongoDB, kafka, microservice)
Тема 13	Патерни відмовостійкості та надійності	1. Definitions 2. Main principles and approaches 3. Identifying failure scenarios 4. Patterns for resiliency - Place for patterns implementation - Application-level - Communication-level - Infrastructure-level
Тема 14-15	Безпека	1. Identity management/Identity and access management for organization 2. Identity management for developers 3. Access control management 4. Authorization approaches 5. Security and architecture styles 6. Standards 7. Multiple organizations (SaaS) 8. Regular users vs service accounts 9. Decentralised identity management
Тема 16	Стратегії випуску та розгалуження (release and branching strategies)	1. Release schedule variations 2. Branching patterns 3. Famous branching strategies (gitflow etc) 4. Selecting branching pattern for your release 5. CI/CD 6. Pipelines 7. Where developer responsibility ends 8. Code vs executable module
Тема 17	Типи збереження даних	1. Hierarchical (1960) 2. Relational (1970) 3. NoSQL: - Key-value - Document - Column-oriented - Graph - Object oriented 4. Purpose-built: - Timeseries - In-memory - Configuration store - Event-sourcing - Full-text search - Geo-spacial - Blob storage 5. Data-lakes 6. Real-time sync & offline work 7. Datastores for modern applications
Тема 18	Кластерні системи. CAP теорема	1. ACID 2. BASE 3. CAP theorem

5.2 Тематика комп'ютерних практикумів/лабораторних робіт

1. Визначення системних вимог
2. Прийняття та документування архітектурних рішень
3. Мікросервісна архітектура
4. Архітектура, заснована на подіях
5. Масштабування
6. Прийняття архітектурних рішень щодо надійності та відмовостійкості системи

6. Самостійна робота студента/аспіранта

Матеріали для самостійного вивчення дисципліни розміщені викладачем в електронному вигляді у Google Classroom, до якого надано доступ групі студентів та асистентам, які ведуть заняття комп'ютерного практикуму. До самостійної роботи студента відноситься, в основному, виконання завдання комп'ютерного практикуму, а також опрацювання лекційного та додаткового теоретичного матеріалу за наданими презентаціями лекцій та додатковою літературою. На самостійну роботу студент має витратити кількість годин, що є втричі більшою за кількість годин, проведених ним на аудиторних заняттях

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Форми організації освітнього процесу, види навчальних занять і оцінювання результатів навчання регламентуються Положенням про організацію освітнього процесу в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/39>) та Положенням про поточний, календарний та семестровий контроль в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/32>).

Студент має вивчати дисципліну протягом семестру, дотримуючись календарного плану виконання завдань комп'ютерного практикуму, вивчення тем лекційного матеріалу та виконання модульних контрольних робіт. Усі завдання студент має виконувати **самостійно і вчасно**. Завдання вважається виконаним, якщо студент захистив завдання комп'ютерного практикуму у викладача. Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити наприкінці кожного лекційного заняття в електронному журналі. Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

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

Поточний контроль: завдання комп'ютерного практикуму, тести

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

Семестровий контроль: сумарна оцінка на базі лабораторних та тестів.

Умови допуску до семестрового контролю: семестровий рейтинг більше 50 балів.

Рейтинг студента з кредитного модуля складається з балів, які він отримує за результатами:

- 1) виконання лабораторних робіт;
- 2) проходження тестів;
- 3) Виконання розрахунково-графічної роботи.

Система рейтингових балів

Лабораторні роботи.

Практичні навички студента оцінюються за результатами захисту виконаних завдань лабораторних робіт.

Тести.

Теоретичні навички студента оцінюються за результатами проходження тестів у Google Classroom.

Підсумкова оцінка формується за результатами оцінювання знань та навичок студента в семестрі за формулою:

$$S=0,7 \cdot Z+0,3 \cdot E$$

де Z – сума оцінок за лабораторні та тести, E – оцінка за РГР.

<i>Кількість балів</i>	<i>Оцінка</i>
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)

Робочу програму навчальної дисципліни (силабус):

Складено асистент кафедри ІІІ, PhD., Смілянець Ф. А.

Ухвалено кафедрою інформатики та програмної інженерії (протокол №16 від 23.06.25)

Погоджено методичною комісією факультету (протокол №11 від 27.06.25)