



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>Денна</i> |
| Рік підготовки, семестр | <i>2 курс, весняний семестр</i> |
| Обсяг дисципліни | <i>4 кредита (120 годин: 36 годин лекцій, 18 годин лабораторних робіт, 66 годин СРС)</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> , <i>асистент Зарічковий О.А.</i> |
| Розміщення курсу | Google Classroom, https://classroom.google.com/c/Nzc1MTg5Mzg4MTYy?cjc=qyzisdms |

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.

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

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

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

1. ФК 03 Здатність розробляти архітектури, модулі та компоненти програмних систем;

2. ФК 07 Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
3. ФК 11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення;
4. ФК 12 Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення;
5. ФК 13 Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
6. ПРН 07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
7. ПРН 08 Вміти розробляти людино-машинний інтерфейс;
8. ПРН 13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
9. ПРН 15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;
10. ПРН 17 Вміти застосовувати методи компонентної розробки програмного забезпечення;
11. ПРН 18 Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

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

При вивченні цієї дисципліни використовуються знання студентів з дисциплін:

- ПО 02 Основи програмування;
- ПО 05 Бази даних;
- ПО 08.1 Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії;
- ПО 08.2 Компоненти програмної інженерії. Частина 2. Моделювання та аналіз вимог до програмного забезпечення.

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. СОА. 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. Матеріали дистанційного курсу.
2. I. Gorton, *Essential Software Architecture*. Springer Berlin Heidelberg, 2011. url:https://www.researchgate.net/publication/220690558_Essential_Software_Architecture_2_ed
3. L. Bass, P. Clements and R. Kazman, *Software Architecture in Practice*, Second Edition. Addison-Wesley Professional, 2003. url: <https://people.ece.ubc.ca/matei/EECE417/BASS/index.html>.

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 лекції | Перелік основних питань | Кількість ауд. годин |
|---------|---|--|----------------------|
| Тема 1 | Поняття архітектури, її місце в циклі розробки програмного забезпечення | <ol style="list-style-type: none"> 1. Definition and purpose 2. Architecture in SDLC. 3. Architecture in preparing sprints in Agile/Scrum. 4. Organizing architecture-related communication. | 2 |
| Тема 2 | Документування архітектури | <ol style="list-style-type: none"> 1. Documenting architecture. 2. Architecture document template. Simplest architecture document example (with system decomposition and API design). | 2 |
| Тема 3 | Архітектурні точки зору (architecture view) | <ol style="list-style-type: none"> 1. C4 views 2. 4+1 views 3. Module views, component-and-connector views, allocation views 4. Notations to present software architecture. | 2 |
| Тема 4 | Якісні атрибути(нефункціональні вимоги) | <ol style="list-style-type: none"> 1. Quality attributes 2. Quality attribute scenarios 3. Availability 4. Performance 5. Security 6. Modifiability etc | 2 |
| Тема 5 | Архітектурі тактики та патерни | <ol style="list-style-type: none"> 1. Tactics 2. Patterns 3. Pattern “Layers” 4. Other patterns | 2 |
| Тема 6 | Моноліт. СОА. Мікросервіси | <ol style="list-style-type: none"> 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 | 2 |
| Тема 7 | Мікросервіси | <ol style="list-style-type: none"> 1. Data: migration scripts 2. Data sync between microservices 3. Data aggregation for microservices | 2 |
| Тема 8 | Мікросервіси (продовження) | <ol style="list-style-type: none"> 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 | 2 |
| Тема 9 | Подійно-орієнтована архітектура | <ol style="list-style-type: none"> 1. Event driven architecture concept 2. Event-driven architecture patterns <ul style="list-style-type: none"> - Event sourcing - Streaming analytics - CQRS - Broker vs Mediator | 2 |
| Тема 10 | Безсерверна архітектура | <ol style="list-style-type: none"> 1. What is serverless? Why serverless? Serverless vs microservices | 2 |

| | | | |
|------------|--|---|---|
| | | <ol style="list-style-type: none"> 2. Use cases 3. Patterns 4. Considerations 5. Serverless Framework | |
| Тема 11 | Простір-орієнтована архітектура, Кешування | <ol style="list-style-type: none"> 1. Space-based architecture 2. Caching patterns 3. Caching topologies | 2 |
| Тема 12 | Масштабування | <ol style="list-style-type: none"> 1. Scaling cube 2. Vertical vs Horizontal scaling 3. Replication vs sharding. 4. Scaling examples (Redis, PostgreSQL, MongoDB, kafka, microservice) | 2 |
| Тема 13 | Патерни відмовостійкості та надійності | <ol style="list-style-type: none"> 1. Definitions 2. Main principles and approaches 3. Identifying failure scenarios 4. Patterns for resiliency <ul style="list-style-type: none"> - Place for patterns implementation - Application-level - Communication-level - Infrastructure-level | 2 |
| Тема 14-15 | Безпека | <ol style="list-style-type: none"> 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 | 2 |
| Тема 16 | Стратегії випуску та розгалуження (release and branching strategies) | <ol style="list-style-type: none"> 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 | 2 |
| Тема 17 | Типи збереження даних | <ol style="list-style-type: none"> 1. Hierarchical (1960) 2. Relational (1970) 3. NoSQL: <ul style="list-style-type: none"> - Key-value - Document - Column-oriented - Graph - Object oriented 4. Purpose-built: <ul style="list-style-type: none"> - Timeseries - In-memory - Configuration store - Event-sourcing - Full-text search - Geo-spacial - Blob storage | 2 |

| | | | |
|------------|-----------------------------------|--|---|
| | | 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 | 2 |

5.2 Тематика комп'ютерних практикумів/лабораторних робіт

| № з/п | Назва лабораторної роботи | Кількість ауд. годин |
|-------|---|----------------------|
| 1 | Лабораторна робота 1. Формування функціональних вимог. | 3 |
| 2 | Лабораторна робота 2. Виділення архітектурно значущих функціональних вимог. Формування нефункціональних вимог. Документація архітектурних рішень. | 3 |
| 3 | Лабораторна робота 3. Мікросервісна архітектура. | 3 |
| 4 | Лабораторна робота 4. Подійно-орієнтована та безсерверна архітектури. | 3 |
| 5 | Лабораторна робота 5. Масштабування. | 3 |
| 6 | Лабораторна робота 6. Забезпечення відмовостійкості та надійності системи.. | 3 |

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.

| № з/п | Назва теми, що виноситься на самостійне опрацювання | Кількість годин СРС |
|-------|---|---------------------|
| 1 | Визначення архітектурно значущих вимог. | 8 |
| 2 | Методи документування вимог | 8 |
| 3 | Методи документування архітектурних рішень | 8 |
| 4 | Робота з даними в мікросервісних архітектурах | 8 |
| 5 | Динамічне масштабування. | 8 |

| | | |
|---|---|----|
| 6 | Відмовостійкість програмного забезпечення | 10 |
| 7 | Безпека в хмарному програмному забезпеченні | 10 |
| 8 | Вибір сховища даних | 6 |

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, МКР

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

Семестровий контроль: сумарна оцінка на базі лабораторних та МКР.

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 МКР;
- 3) виконання розрахунково-графічної роботи.

Система рейтингових балів

Лабораторні роботи.

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

Критерії оцінювання лабораторних робіт

| Виконання | Захист |
|--|--|
| Робота виконана без зауважень – 7 балів | Робота захищена без зауважень – 3 балів |
| Достатньо повне виконання роботи з деякими похибками – 5 балів | Робота захищена, однак при захисті є зауваження – 2 балів |
| Неповністю виконана робота – 3 бали | Робота захищена, однак на частину питань відсутні відповіді або надані часткові відповіді – 1 бали |
| При виконанні роботи є суттєві зауваження – 0 балів | Є суттєві зауваження при захисті роботи – 0 балів |

Мінімальний бал за кожну роботу – 6 (60% від максимуму) або 0 у випадку неналежного виконання.

Модульні контрольні роботи.

Протягом семестру передбачено проведення 4 модульних контрольних робіт за темами, котрі були розглянуті при вивченні освітнього компоненту. Максимальний сумарний бал за даний контрольний захід – 20 балів:

| Тест | Максимальна оцінка |
|------|--------------------|
| 1 | 11 балів |
| 2 | 8 балів |
| 3 | 4 балів |
| 4 | 8 бали |
| 5 | 7 балів |
| 6 | 4 бали |

Бал визначається кількістю правильних наданих відповідей на тестові питання.

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.

Необхідною умовою допуску до семестрового контролю є:

- виконана МКР на позитивну оцінку;

- виконано та зараховано цикл лабораторних робіт на позитивну оцінку;
- виконана РГР на позитивну оцінку.

Студенти, які наприкінці семестру мають рейтинг менше 60 балів і є допущеними до семестрового контролю, а також ті, хто хоче підвищити оцінку, виконують залікову контрольну роботу. При цьому до існуючих балів додаються бали за контрольну роботу і ця рейтингова оцінка є остаточною. Завдання залікової контрольної роботи складаються з одного практичного завдання з різних розділів дисципліни.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

| <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., асистент Зарічковий О.А.

Ухвалено кафедрою інформатики та програмної інженерії (протокол №2/1 від 10.10.2025)

Погоджено методичною комісією факультету (протокол №3 від 17.10.2025)