Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра інженерії програмного забезпечення в енергетиці

Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення

Силабус освітнього компонента

Реквізити навчальної дисципліни

Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*12 «Інформаційні технології»*
Спеціальність	*121 «Інженерія програмного забезпечення»*
Освітня програма	*Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці*
Статус дисципліни	*Обов'язкова (нормативна)*
Форма навчання	*Очна (денна)*
Рік підготовки, семестр	*II курс, весняний семестр*
Обсяг дисципліни	*5 кредитів ECTS /150 годин (36 годин лекцій, 36 годин практичних занять (комп'ютерних практикумів))*
Семестровий контроль/ контрольні заходи	*Екзамен/тестування, МКР, захист лабораторних робіт*
Розклад занять	*1 лекція (2 години) 1 раз на тиждень;* *1 практичне заняття (2 години) 1 раз тиждень.*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Лектор: *к.т.н.Варава Іван Андрійович,* Практичні заняття: *к.т.н.Варава Іван Андрійович,*
Розміщення курсу	*Кампус*

Програма навчальної дисципліни

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

Силабус освітнього компонента «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення» складено відповідно до освітньої програми підготовки бакалаврів «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 – Інженерія програмного забезпечення.

***Метою навчальної дисципліни** є формування та закріплення у студентів наступних компетентностей:*

(ФК1) Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення; (ФК2) Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; (ФК3) Здатність розробляти архітектури, модулі та компоненти програмних систем; (ФК4) Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами; (ФК5) Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу; (ФК7)Володіння знаннями про інформаційні моделі даних, здатність створювати

програмне забезпечення для зберігання, видобування та опрацювання даних; (ФК8) Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення; (ФК10) Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення тестування і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя; (ФК11) Здатність реалізувати фази та ітерації життєвого циклу програмних систем інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; (ФК 12) Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення; (ФК 13) Здатність обґрунтовано обирати та освоювати інструментарій з розробки тестування та супроводження програмного забезпечення.

***Предмет навчальної дисципліни** – методи проектування програмного забезпечення, принципи та підходи до вибору технологій структурування додатків, управління даними, масштабування та забезпечення безпеки.*

Програмні результати навчання, на формування та покращення яких спрямована дисципліна:

(ПРН1) Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; (ПРН2) Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності; (ПРН3) Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення; (ПРН4) Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення; (ПРН6) Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; (ПРН7) Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення; (ПРН 8) Вміти розробляти людино-машинний інтерфейс; (ПРН9) Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення; (ПРН10) Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування; (ПРН11) Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; (ПРН13) Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; (ПРН14) Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення; (ПРН15) Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення; (ПРН16) Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; (ПРН17) Вміти застосовувати методи компонентної розробки програмного забезпечення;(ПРН18) Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних; (ПРН19) Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення; (ПРН20) Знати підходи щодо оцінки та забезпечення якості програмного забезпечення; (ПРН26) Вміти використовувати методи інженерії даних.

2. Пререквізити та постреквізити дисципліни

Для успішного засвоєння дисципліни студент повинен володіти освітніми компонентами «Об'єктно-орієнтований аналіз та конструювання програмних систем» (ПО 12). Компетенції, знання та уміння, одержані в процесі вивчення освітнього компонента є необхідними для подальшого вивчення освітніх компонентів «Безпека програмного забезпечення» (ПО 9), «Методологія розробки інтелектуальних комп'ютерних програм» (ПО 19).

3. Зміст навчальної дисципліни

Розділ №1. Вступ до проектування архітектури ПО. Складові архітектури

Тема 1. Поняття архітектури програмного забезпечення. Класифікація та різновиди архітектур.

Тема 1.1. Способи представлення архітектури. Засоби проектування архітектури програмних систем.

Тема 1.2. Способи опису взаємодії з програмним рішенням. Проектування взаємодії користувача з системою.

Тема 1.3. Опис взаємодії зовнішніх та внутрішніх систем, що відносяться до програмного рішення.

Розділ №2.

Тема 2. Варіанти архітектурних рішень та принципи зв'язку компонентів програмних систем

Тема 2.1. Базові принципи передачі інформації. Шина повідомлень.

Тема 2.2. Патерни програмування програмного забезпечення

Тема 2.3. Поєднання принципів розробки із шаблонами розробки

Тема 2.4. Антипатерни та недоліки надмірних обмежень через хибне застосування шаблонів

Тема 2.5. Принцип MONAD

Розділ №3. Різновиди архітектурних моделей. Застосування

Тема 3. Архітектурні рівні. Атомарність моделювання архітектури систем.

Тема 3.1. Монолітна архітектура додатка, сервісу; переваги та недоліки

Тема 3.2. Мікросервісна архітектура програмного рішення, її переваги та недоліки

Тема 3.3: Багатошарова архітектура та її застосування для створення бізнес рішень

Тема 3.4. Різнорівневі архітектури, складність зв'язності елементів, що вони описують

Тема 3.5. Сервіс-орієнтована архітектура. Проектування архітектури із врахуванням взаємодії з існуючими рішеннями. Клієнт-серверні системи.

Розділ №4. Засоби проектування інтеграції програмного продукту та його життєвого циклу

Тема 4. Вплив проектування архітектури системи на взаємодію користувача з системою. Засоби опису програмного рішення. Засоби створення документації (Obsidian).

Тема 5. Розширюваність систем. Вплив різних підходів в проектування архітектури на легкість, швидкість та якість розширення програмних рішень.

4. Навчальні матеріали та ресурси

Основна література

1. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.

2. Martin Fowler Patterns of Enterprise Application Architecture 1st Edition, Addison-Wesley Professional, 2002, 560 p.

3. Sam Newman Building Microservices O'Reilly Media, 2015. – 280p.

4. Len Bass Software Architecture in Practice Addison-Wesley Professional, 4th Edition, 2022, 442 p.

5. Craig Larman Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Ed, Prentice Hall.2004. – 736p.

Додаткова література

6. *Craig Larman Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Prentice Hall.2004. – 736p.*
7. *Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software / Addison-Wesley Professional., 1995. – 396 p.*
8. *Martin R. Clean Architecture: A Craftsman\'s Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition / Robert C. Martin., 2018. – 420p.*
9. *Martin Fowler. Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2003. – 560p.*
10. *Eric Evans. Domain-Driven Design: Tackling Complexity in the Heart of Software Addison-Wesley Professional, 2004. – 534p.*
11. *Albin S. The Art of Software Architecture Design Methods and Techniques / Stephen T. Albin., 2003. – 336 с.*
12. *Docs|MDX [Електронний ресурс] – Режим доступу https://mdxjs.com/docs/*

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела)
1	***Тема 1. Поняття архітектури програмного забезпечення. Класифікація та різновиди архітектур.*** *Основні питання: Причина необхідності створення моделі архітектури програмного забезпечення. Різновиди систем та загальновідомі стійкі рішення.*
2	***Тема 1.1. Способи представлення архітектури. Засоби проектування архітектури програмних систем.*** *Основні питання: різновиди та сфери застосування діаграм, що входять до складу загальної архітектури. Опис проектування різних елементів за допомогою спеціалізованих діаграм.*
3	***Тема 1.2. Способи опису взаємодії з програмним рішенням. Проектування взаємодії користувача з системою.*** *Основні питання: взаємодія користувача з додатком. Різновиди керування та отримання інформації в додатку.*
4	***Тема 1.3. Опис взаємодії зовнішніх та внутрішніх систем, що відносяться до програмного рішення.*** *Основні питання: порядок взаємодії компонентів. Час існування запиту до системи. Різновиди способів обробки запитів. Взаємодія із зовнішніми сервісами*
5	***Тема 2. Варіанти архітектурних рішень та принципи зв'язку компонентів програмних систем*** *Основні питання: Ідея монолітної архітектури, її переваги у швидкості розробки, легкості проектування до стану MVP та для яких бізнес рішень може бути використана*
6	***Тема 2.1. Базові принципи передачі інформації. Шина повідомлень.*** *Основні питання: різновиди передачі інформації між елементами системи та систем. Види інформації та способи її конвертації*
7	***Тема 2.2. Патерни програмування програмного забезпечення*** *Основні питання: Патерни, їх види та сфери застосування*
8	***Тема 2.3. Поєднання принципів розробки із шаблонами розробки*** *Основні питання: Приципи та підходи до розробки: (SOLID, KISS, WET, DRY). Поєднання принципів із шаблонами/патернами розробки*
9	***Тема 2.4. Антипатерни та недоліки надмірних обмежень через хибне застосування шаблонів*** *Основні питання: невірні імплементації шаблонів. Напівшаблонізовані системи. Перепроектовані системи. Накладання стилів*
10	***Тема 2.5. Принцип MONAD*** *Основні питання: основоположення приниципу MONAD. Його вплив на швидкість та якість розробки компонентів та сервісів.*
11	***Тема 3. Архітектурні рівні. Атомарність моделювання архітектури систем.*** *Основні питання: для яких елементів системи програмного рішення необхідно спроектувати архітектуру та які елементи взаємодіючих компонентів ПО підлягають опису*
12	***Тема 3.1. Монолітна архітектура додатка, сервісу; переваги та недоліки***

	Основні питання: Ідея монолітної архітектури, її переваги у швидкості розробки, легкості проектування до стану MVP та для яких бізнес рішень може бути використана
13	***Тема 3.2. Мікросервісна архітектура програмного рішення, її переваги та недоліки*** *Основні питання: поділ системи на елементи з різною відповідальністю. Проектування зв'язку елементів системи. Розподіл системних потужностей для різних елементів системи*
14	***Тема 3.3: Багатошарова архітектура та її застосування для створення бізнес рішень*** *Основні питання: поняття шарів представлення системи. Опис типових шарів для взаємодії із системою. Проектування способів взаємодії шарів системи. Обмеження каналів зв'язку між шарами.*
15	***Тема 3.4. Різнорівневі архітектури, складність зв'язності елементів, що вони описують.*** *Основні поняття: Поняття рівнів систем. Причини існування різних рівнів систем. Способи взаємодії елементів системи або систем на різних рівнях*
16	***Тема 3.5. Сервіс-орієнтована архітектура. Проектування архітектури із врахуванням взаємодії з існуючими рішеннями. Клієнт-серверні системи.*** *Основні питання: Принципи та причини існування сервіс-орієнтованого підходу до проектування архітектури. Основи та приклади опису взаємодії додатків. Поняття CORS-політики. Клієнт-серверні системи. Товстий клієнт. Тонкий клієнт. Одноланкова система. Дволанкова система.*
17	***Тема 4. Вплив проектування архітектури системи на взаємодію користувача з системою. Засоби опису програмного рішення. Засоби створення документації.*** *Основні питання: Вплив проектування архітектури системи на взаємодію користувача з системою. Способи представлення та передачі документації. Формування контекстної документації. Формування сервісної документації*
18	***Тема 5. Розширюваність систем. Вплив різних підходів в проектування архітектури на легкість, швидкість та якість розширення програмних рішень.*** *Основні питання: Що таке розширення системи? Коли настає необхідність її розширювати? Які способи розширення систем існуються (горизонтальний, вертикальний, діагональний). Чим відрізняються способи розширення системи та як проектування архітектури пов'язане з цим?*

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	***Практичне заняття № 1. Вступ до поняття програмної архітектури*** *Основні питання: поняття архітектури системи, принцип проектування перед розробкою. Елементи, що підлягають проектуванню.*
2	***Практичне заняття № 2. Класифікація програмної архітектури*** *Основні питання: ідентифікаційні ознаки архітектур.*
3	***Практичне заняття № 3. Засоби представлення архітектури*** *Основні питання: Різновиди засобів для представлення різних компонент архітектури.*
4	***Практичне заняття № 4. Проектування взаємодії користувача із системою*** *Основні питання: способи представлення взаємодії користувача із системою, різновиди користувачів, рівні доступу, проектування обмежень та штрафів.*
5	***Практичне заняття № 5. Проектування взаємодії зовнішніх та внутрішніх компонентів систем***

	Основні питання: ролі компонентів в системі, взаємодія компонентів різних ролей, способи представлення взаємодії компонент, проектування передачі даних між компонентами.
6	***Практичне заняття № 6. Потік інформації між компонентами*** *Основні питання: Способи передачі інформації. Види запитів. Лімітовані канали.*
7	***Практичне заняття № 7. Модульна контрольна робота №1. Проектування взаємодії користувача з програмним рішенням*** *Основні питання: UML діаграма. Діаграма класів. Види запитів.*
8	***Практичне заняття № 8. Різновид архітектурних рішень. Загальні положення складу архітектури. Формування стилю розробки програмного забезпечення.*** *Основні питання: причини існування різних архітектурних рішень. Проектування динамічної та статичної системи. Патерни, принципи розробки програмного забезпечення. Сфери використання різних підходів та проектування архітектури з урахуванням обмежень шаблонів та принципів.*
9	***Практичне заняття № 9. Антипатерни та комбінації шаблонів*** *Основні питання: вплив технічного завдання на проектування архітектури. Проектування обмеження можливостей програмного продукту.*
10	***Практичне заняття № 10. Використання MONAD для формування абстрактних шарів розробки*** *Основні питання: MONAD підхід для проектування та розробки програмного забезпеченні. Переваги та недоліки додаткових шарів абстракції.*
11	***Практичне заняття № 11. Проектування складу компонент програмного рішення*** *Основні питання: неподільність компонент. Уніфікація рішень в рамках сервісу та системи .*
12	***Практичне заняття № 12. Проектування монолітного програмного рішення*** *Основні питання: складові монолітної архітектури. Забезпечення життєздатності та відмовостійкості внутрішніх компонент програмного рішення.*
13	***Практичне заняття № 13. Проектування мікро сервісного програмного рішення*** *Основні питання: складові мікро сервісної архітектури. Забезпечення зв'язку компонент. Поняття "лінивих" компонент. Динамічний та статичний цикл життя компонент.*
14	***Практичне заняття № 14. Проектування багатошарового програмного рішення*** *Основні питання: врахування потреб користувача. Створення шарів комунікації компонент. Створення спеціалізованих шарів для взаємодії користувача або сервісу із програмним рішенням*
15	***Практичне заняття № 15 Проектування архітектури рівнів програмного рішення*** *Основні питання: рівні програмного рішення. Взаємодія та передача інформації між рівнями. Переваги та недоліки багаторівневої системи. Проектування інтерфейсів та каналів передачі інформації*
16	***Практичне заняття № 16. Проектування сервіс-орієнтованої архітектури*** *Основні питання: Проектування взаємодії програмного рішення зі сторонньою системою. Ідентифікація обмежень сторонньої системи. Проектування адаптерів та шарів для комунікації зі сторонніми системами. Уніфікація каналу зв'язку програмних рішень під час паралельного проектування. Вплив проектування архітектури системи на взаємодію користувача з системою.*
17	***Практичне заняття № 17. Створення документації до програмного забезпечення. Розширення можливостей системи. Масштабування систем.***

	Основні питання: засоби створення документації, компонентне розширення системи. Взаємодія розширювальних компонентів.
18	***Практичне заняття № 18. Модульна контрольна робота №2. Проектування багатошарового сервіс-орієнтованого рішення*** *Основні питання: Діаграма компонентів. Діаграма послідовностей. Діаграма зв'язку компонентів системи. Адаптери.*

Самостійна робота студента

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	*Підготовка до практичних занять*	*36*
2	*Підготовка до МКР*	*12*
3	*Підготовка до екзамену*	*30*

6. Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного вирішення задач та складанні та компіляції програм.

Модульна контрольна робота (МКР) виконується після вивчення Розділів 1-3 та виконання практичних занять 1-5. Контрольна робота проводяться у вигляді тестування за допомогою Google Forms.

Політика та контроль

6. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- *правила відвідування занять: заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.*
- *правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;*
- *політика дедлайнів та перескладань: якщо студент не проходив або не з'явився на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;*
- *політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» https://kpi.ua/files/honorcode.pdf встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Компоненти програмної інженерії. Частина 4. Архітектура програмного забезпечення»;*
- *при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.*

7. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

***Поточний контроль**: виконання завдань до практичних занять, МКР.*
***Календарний контроль**: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.*
Семестровий контроль: *екзамен.*
***Умови допуску до семестрового контролю**: виконані та захищені практичні роботи, виконані завдання до практичних занять, семестровий рейтинг більше 40 балів.*
Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	*Оцінка*
95-100	*Відмінно*
85-94	*Дуже добре*
75-84	*Добре*
65-74	*Задовільно*
60-64	*Достатньо*
Менше 60	*Незадовільно*
Менше 30	*Не допущено*

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- *тестування по кожному лекційному заняттi;*
- *виконання та захист практичних робіт;*
- *виконання модульної контрольної роботи (МКР);*
- *відповіді на екзамені.*

Тестування по лекціям	*Практичні роботи*	*МКР*	*Екзамен*
9	*54*	*7*	*30*

Тестування по матеріалам лекційних занять

***Ваговий бал 1.** Максимальна кількість балів за тестування – 0,5 балу * 18 лекцій = 9 балів.*

Тестування проводиться за допомогою Google Forms наприкінці поточної лекції. Тривалість проходження одного тестування – 5 хвилин. Кількість спроб – одна. У деяких випадках, що пов'язані з технічними проблемами студентів, може надатися повторна спроба на окремі тестування.

Кожне тестування містить 5 запитань різного формату (вибір одного правильного варіанту з переліку; вибір декількох правильних варіантів з переліку; визначити відповідність, розгорнута відповідь).

Критерії оцінювання

- *запитання типу «вибір правильного варіанту з переліку», «вибір декількох правильних варіантів з переліку» оцінюються однозначно: вірна відповідь – 0,1 бал, невірна відповідь – 0 балів;*
- *запитання типу «визначити відповідність» оцінюються у відповідності до кількості елементів у тесті відповідно: жодної вірної відповіді – 0 балів, частково вірна відповідь – 0,01-0,09 балів, вірна відповідь 0,1 бал.*
- *запитання типу «розгорнута відповідь» оцінюється віл 0 до 0,1 балу в залежності від точності формулювання, повноти відповіді.*

Практичні заняття

Ваговий бал 1*. Максимальна кількість балів за всі практичні заняття – 3 бали * 18 занять= 54 балів.*

На практичних занять студенти виконують практичні роботи за тематикою практичного заняття. Після кожного практичного заняття студенти отримують домашнє завдання, яке необхідно вирішити та надати на перевірку викладачу до початку наступного заняття, як правило, на наступному тижні.

Критерії оцінювання

- домашнє завдання вирішено вірно та здано протягом тижня після практичного заняття – 3 бал;

- домашнє завдання вирішено із незначними помилками та здано протягом тижня – 2,5 бали;

- домашнє завдання вирішено із незначними помилками та здано протягом більш ніж одного тижня після практичного заняття – 2 балів;

- домашнє завдання вирішено із значними помилками – повертається на доопрацювання.

Модульна контрольна робота

Ваговий бал – 7. *Модульна контрольна робота (МКР) виконується під час календарного контролю. Модульна контрольна робота розділена на дві частини у вигляді тестів GoogleForms.*

Критерії оцінювання модульної контрольної роботи:

На модульній контрольній роботі студент відповідає на 20 запитань. Кожне запитання оцінюється аналогічно критеріям оцінки тестування за матеріалами лекційних занять з коефіцієнтом 10. Таким чином рейтинговий бал за МКР складає:

$$R_{МКР} = (7/2_{мкр}) *2_{мкр}* 10_{зп}*0,1_{бал} = 7 \text{ балів}$$

Календарний контроль

Календарний контроль базується на поточній рейтинговій оцінці. Умовою позитивної атестації є значення поточного рейтингу студента не менше 50% від максимально можливого на час атестації. Бал, необхідний для отримання позитивного календарного контролю доводиться до відома студентів викладачем не пізніше ніж за 2 тижні до початку календарного контролю.

Додаткові (бонусні) бали

Рейтинговою системою оцінювання передбачені додаткові бали за виконання додаткових завдань. Один студент не може отримати більше ніж 5 бонусних балів у семестрі. Бонусні бали можуть бути отримані за активну участь під час лекцій (відповіді на уточнюючі питання, приведення прикладів) та/або участь у підготовці тез конференцій.

Форма семестрового контролю – екзамен

Максимальна сума балів за роботу у семестрі складає 70. Необхідною умовою допуску до екзамену виконані завдання до практичних занять, семестровий рейтинг не менше 40 балів.

Екзамен містить дві складові: теоретичну та практичну. ***Теоретична складова*** *направлена на перевірку набутих в результаті вивчення освітнього компонента знань студентів у вигляді розгорнутих відповідей на два питання за лекційним матеріалом семестру. Максимальна кількість балів за теоретичні питання складає: 2 питання * 10*

*балів = 20 балів. **Практична складова** передбачає перевірку набутими студентами умінь розробляти архітектуру програмних систем, застосовувати інструментальні засоби опису архітектури програмного забезпечення. Кожному студенту надається окрема задача, відповідно до умов якої необхідно розробити архітектуру програмної системи для заданої предметної області. Максимальна кількість балів за задачу складає 10 балів.*

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль (екзамен):

1. Поняття архітектури програмного забезпечення.
2. Класифікація архітектур програмного забезпечення.
3. Способи представлення архітектур.
4. Поняття Enterprise-архітектури
5. Монолітна архітектура
6. Клієнт-серверна архітектура
7. Сервіс-орієнтована архітектура
8. Багатошарова архітектура
9. Мікросервісна архітектура
10. Чиста архітектура
11. Контейнеризація програмного забезпечення
12. Впровадження залежностей (Dependency injection)
13. Визначення поняття патерну програмування
14. Визначення поняття стиль програмування
15. Визначення поняття принцип програмування
16. Застосування UML діаграм в проектуванні програмного забезпечення
17. Діаграма класів та її застосування
18. Діаграма компонентів та її застосування
19. Багаторівнева архітектура
20. Способи масштабування системи
21. Визначення поняття рівні абстракції програмного забезпечення
22. Діаграма послідовностей та її застосування
23. Визначення патерну Фабрика
24. Визначення патерну Будівельник
25. Визначення патерну Наглядач
26. Визначення патерну Ланцюжок залежностей
27. Шаблон MVC
28. Шаблон MVVM
29. Feature Sliced Design
30. Яка роль архітектури програмного забезпечення у процесі розробки програмних продуктів?
31. Які причини потреби в створенні моделі архітектури програмного забезпечення?

32. Які різновиди систем існують та як вони впливають на архітектуру програмного забезпечення?
33. Які загальновідомі стійкі рішення в архітектурі програмного забезпечення?
34. Які різновиди діаграм використовуються для представлення архітектури програмних систем?
35. Для яких цілей застосовуються конкретні типи діаграм в загальній архітектурі?
36. Яким чином спеціалізовані діаграми описують різні елементи архітектури програмних систем?
37. Які основні аспекти взаємодії користувача з програмними додатками?
38. Які різновиди керування та отримання інформації в програмному додатку існують?
39. Як проходить взаємодія між компонентами в системі програмного рішення?
40. Які різновиди способів обробки запитів використовуються в архітектурі програмного забезпечення?
41. Як відбувається взаємодія зовнішніх сервісів у програмному рішенні?
42. Яка ідея монолітної архітектури та її переваги в розробці програмних продуктів?
43. Для яких бізнес-рішень може бути використана монолітна архітектура?
44. Які різновиди передачі інформації між елементами системи та системами існують?
45. Які види інформації і способи її конвертації використовуються в архітектурі програмного забезпечення?
46. Що таке патерни програмування, які існують види патернів та в яких сферах вони застосовуються?
47. Які принципи розробки існують і як вони поєднуються з шаблонами розробки?
48. Які підходи до розробки використовують ці принципи і шаблони?
49. Які можуть бути наслідки неправильної імплементації шаблонів у програмному рішенні?
50. Які приклади недоліків можуть бути у системах через перепроектованість та накладання стилів?
51. Яке основне призначення принципу MONAD?
52. Як він впливає на швидкість і якість розробки компонентів та сервісів?
53. Які елементи програмного рішення потребують проектування архітектури?
54. Які елементи взаємодіючих компонентів програмного забезпечення підлягають опису?
55. Які переваги та недоліки монолітної архітектури у швидкості розробки?
56. Для яких бізнес-рішень може бути використана монолітна архітектура?
57. Як відбувається поділ системи на елементи з різною відповідальністю в мікросервісній архітектурі?
58. Як розподіляються системні потужності для різних елементів системи у мікросервісній архітектурі?
59. Яке поняття шарів представлення системи використовується в багатошаровій архітектурі?
60. Які типові шари використовуються для взаємодії із системою у багатошаровій архітектурі?
61. Які поняття рівнів систем використовуються у різнорівневих архітектурах?
62. Які способи взаємодії елементів системи на різних рівнях існують?

63. Які принципи та причини існування сервіс-орієнтованого підходу до проектування архітектури?
64. Як відбувається взаємодія додатків у сервіс-орієнтованій архітектурі?
65. Як впливає проектування архітектури на взаємодію користувача з системою?
66. Які засоби представлення та передачі документації використовуються для опису програмного рішення?
67. Що означає розширення системи та коли настає необхідність її розширювати?
68. Які способи розширення систем існують та в чому їх суть?

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри інженерії програмного забезпечення, к.т.н. Варавою Іваном Андрійовичем.

Ухвалено кафедрою інженерії програмного забезпечення (протокол № 28 від 28.05.2023 р.)

Погоджено Методичною комісією НН ІАТЕ (протокол № 9 від 26.05.2023 р.)