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 кіберфізичних систем в енергетиці*
Статус дисципліни	*Обов'язкова (нормативна)*
Форма навчання	*Заочна*
Рік підготовки, семестр	*II курс, весняний семестр*
Обсяг дисципліни	*5 кредитів ECTS /150 годин (4 годин лекцій, 2 годин практичних занять (комп'ютерних практикумів),СРС 144*
Семестровий контроль/ контрольні заходи	*Екзамен/МКР, захист практичних робіт*
Розклад занять	*2 лекції (4 години) згідно розкладу;* *1 практичне заняття (2 години) згідно розкладу.*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доц. Шуклін Герман Вікторович, mathacadem-kiev@ukr.net (у робочий час) Практичні: к.т.н., доц. Шуклін Герман Вікторович, mathacadem-kiev@ukr.net (у робочий час)
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 кіберфізичних систем в енергетиці» спеціальності 121 – Інженерія програмного забезпечення.

***Метою навчальної дисципліни** є формування та закріплення у студентів наступних компетентностей: (ФК3) Здатність розробляти архітектури, модулі та компоненти програмних систем; (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; (ФК 12) Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності 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:

(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; (ПРН13) Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; (ПРН17) Вміти застосовувати методи компонентної розробки програмного забезпечення.

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 програмного забезпечення. Складові архітектури

Тема 1.1. Поняття архітектури програмного забезпечення. Класифікація та різновиди архітектур.

Тема 1.2. Способи представлення архітектури. Засоби проектування архітектури програмних систем.

Тема 1.3. Способи опису взаємодії з програмним рішенням. Проектування взаємодії користувача з системою.

Тема 1.4. Опис взаємодії зовнішніх та внутрішніх систем, що відносяться до програмного рішення.

Розділ №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

Розділ №3. Різновиди архітектурних моделей. Застосування

Тема 3.1 Архітектурні рівні. Атомарність моделювання архітектури систем.

Тема 3.2. Монолітна архітектура додатка, сервісу; переваги та недоліки

Тема 3.3. Мікросервісна архітектура програмного рішення, її переваги та недоліки

Тема 3.4: Багатошарова архітектура та її застосування для створення бізнес рішень

Тема 3.5. Різнорівневі архітектури, складність зв'язності елементів, що вони описують

Тема 3.6. 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 4.1 Вплив проектування архітектури системи на взаємодію користувача з системою. Засоби опису програмного рішення. Засоби створення документації.

Тема 4.2 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, AddisonWesley 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

7. *Craig Larman Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Prentice Hall.2004. – 736p.*
8. *Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software / Addison-Wesley Professional., 1995. – 396 p.*
9. *Martin R. Clean Architecture: A Craftsman\'s Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition / Robert C. Martin., 2018. – 420p.*
10. *Martin Fowler. Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2003. – 560p.*
11. *Eric Evans. Domain-Driven Design: Tackling Complexity in the Heart of Software AddisonWesley Professional, 2004. – 534p.*
12. *Albin S. The Art of Software Architecture Design Methods and Techniques / Stephen T. Albin., 2003. – 336 с.*
13. *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 2. 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.*

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	***Практичне заняття № 1. Проектування триланкової архітектури програмної системи*** *Основні питання: поняття архітектури системи; шари бізнес-логіки, доступу до даних, представлення; діаграми UML; поняття сервера, сервера додатків, сервера бази даних та клієнта.*

Самостійна робота студента

1	***Розділ №1. Вступ до проектування архітектури програмного забезпечення. Складові архітектури***
	Тема 1.3. *Способи опису взаємодії з програмним рішенням. Проектування взаємодії користувача з системою.* ***Тема 1.4.*** *Опис взаємодії зовнішніх та внутрішніх систем, що відносяться до програмного рішення.*
2	***Розділ №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*
3	***Розділ №3. Різновиди архітектурних моделей. Застосування***
	Тема 3.1 *Архітектурні рівні. Атомарність моделювання архітектури систем.* ***Тема 3.2.*** *Монолітна архітектура додатка, сервісу; переваги та недоліки* ***Тема 3.3.*** *Мікросервісна архітектура програмного рішення, її переваги та недоліки*
	Тема 3.4. *Багатошарова архітектура та її застосування для створення бізнес рішень* ***Тема 3.5.*** *Різнорівневі архітектури, складність зв'язності елементів, що вони описують* ***Тема 3.6.*** *Сервіс-орієнтована архітектура. Проектування архітектури із врахуванням взаємодії з існуючими рішеннями. Клієнт-серверні системи.*
4	***Розділ №4. Засоби проектування інтеграції програмного продукту та його життєвого циклу***
	Тема 4.1 *Вплив проектування архітектури системи на взаємодію користувача з системою. Засоби опису програмного рішення. Засоби створення документації.* ***Тема 4.2*** *Розширюваність систем. Вплив різних підходів в проектування архітектури на легкість, швидкість та якість розширення програмних рішень.*

6. Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного проектування архітектури програмного забезпечення.

Модульна контрольна робота (МКР) виконується у вигляді тестування за допомогою Google Forms під час залікового тижня.

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.*
***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:

- *виконання та захист практичної роботи (30 балів); -*
  виконання модульної контрольної роботи (МКР) (30 балів); -
  відповіді на екзамені (40 балів).

Форма семестрового контролю – екзамен

Максимальна сума балів за роботу у семестрі складає 60. Необхідною умовою допуску до екзамену виконані завдання до практичного заняття, 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 * 13 балів = 26 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 14 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 в енергетиці, к.т.н. Варавою І.А.*

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

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