

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення



Декан інформаційних технологій

Тетяна ГОВОРУЩЕНКО

Ім'я, ПРІЗВИЩЕ

2024 р.

СІЛАБУС

Навчальна дисципліна Реінжиніринг та оптимізація програмних систем

Освітньо-професійна програма інженерія програмного забезпечення

Рівень вищої освіти другий (магістерський)

Загальна інформація

Позиція	Зміст інформації
Викладач(і)	Форкун Юрій Вікторович
Профайл викладача	https://ipz.khmnu.edu.ua/forkun-yu-v/
Е-mail викладача(ів)	forkunyu@khmnu.edu.ua
Контактний телефон	
Сторінка дисципліни в ІСУ	https://msn.khnu.km.ua/course/view.php?id=6767
Навчальний рік	2024-2025
Консультації	Очні: понеділок, 2-га пара, 1-202 онлайн: за необхідністю та попередньою домовленістю (Zoom. ID: 554 822 9369)

Характеристика дисципліни

Статус	Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
				Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота слухача	Самостійна робота, в т.ч. ІРС	Курсовий проект	Курсова робота	Залік	Іспит
						Разом	Лекції	Лабораторні роботи	Практичні заняття						
О	Д	1	1	5	150	51	17	17	17	99					+

Анотація дисципліни

Дисципліна «Реінжиніринг та оптимізація програмних систем тем» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення»

Мета і завдання дисципліни

Мета дисципліни: формування компетентностей, необхідних для абстрактного мислення, аналізу та синтезу на відповідних рівнях оптимізації проектування та розробки програмних систем та їх компонентів; опанування студентами компонентних технологій та інтегрованих середовищ розробки для ефективної оптимізації програмного коду та програмних систем, конкретних виробничих задач інженерії програмного забезпечення.

Предмет дисципліни: Предметом вивчення дисципліни є реінжиніринг та оптимізація програмних систем.

Завдання дисципліни: Надати студентам знання і практичні навички практичного вирішення прикладних задач з обґрунтуванням реінжинірингу та оптимізації програмних систем

Очікувані результати навчання

Студент, який успішно завершив вивчення дисципліни, має: Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу. Прогнозувати розвиток програмних систем та інформаційних технологій. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника. Планувати, організувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. Вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосуванням машинного навчання.

Тематичний і календарний план вивчення дисципліни

№ тижня	Тема лекції*	Тема лабораторної роботи *	Тема практичної роботи	Самостійна робота студентів		
				Зміст	Год.	Література
1	2	3	4	5	6	7
1	Продуктивність, якість та оптимізація програмних систем. Програмні та організаційні заходи визначення продуктивності та оптимізації ПЗ. Оптимізація етапів процесу розробки, їх закони, правила та умови проектування. Літ.: [1-6]	Лабораторна робота №1. Частина 1. Вимоги до проєктів. Специфікації завдань. Умови оптимізації. Літ.: [1]	Практична робота №1. Процеси розробки, їх закони, правила та умови проектування. Літ.: [1-4]	Опрацювання конспекту лекцій Т1	5	[1; 2;3]
2	Основи реінжинірингу програмного забезпечення. Етапи та процеси реінжинірингу програмного забезпечення. Основні проблеми при плануванні задач реінжинірингу. Оптимізація робіт на етапі проектування програмної системи. Оцінка об'єму робіт на етапі проектування. Зворотня розробка програмного забезпечення. Основні задачі та етапи зворотньої розробки. Ризики повторного використання коду програмного забезпечення. Літ.: [1-6]	Лабораторна робота №1. Частина 2. Вимоги до проєктів. Специфікації завдань. Умови оптимізації. Літ.: [1]	Практична робота №2. Процеси розробки, їх закони, правила та умови проектування. Літ.: [1-4]	Опрацювання конспекту лекцій Т1; підготовка до лабораторного заняття	10	[1; 3]
3	Аналіз архітектури програмного забезпечення. Вимоги до проєктів. Специфікації завдань. Аналіз знань та	Лабораторна робота №2. Частина 1. Реінженерія архітектури програмного	Практична робота №3. Типи архітектур. Типи	Опрацювання конспекту лекцій Т2	10	[2; 4]

	алгоритмів. Специфікації завдань. Типи архітектур та їх моделі. Архітектури, засновані на потоках даних. Незалежні компоненти. Віртуальні машини. Репозиторні архітектури. Рівневі архітектури. Додатки зі змішаною архітектурою. Процедура вибору архітектури. Типи кодування. Літ.: [4,10]	забезпечення. Літ.: [4,12]	кодування. [4,10]			
4	Рефакторінг та оптимізаційні перетворення програмного забезпечення. Принципи рефакторингу. Формат методів рефакторингу. Організація та оптимізація програмного коду. Організація та оптимізація даних. Літ.: [1,6]	Лабораторна робота №2. Частина 2. Реінженерія архітектури програмного забезпечення. Літ.: [4,10]	Практична робота №4. Типи архітектур. Типи кодування. [4,10]	Опрацювання конспекту лекцій Т3; підготовка до лабораторного заняття	10	[2; 5]
5	Організація та оптимізація алгоритмів. Інструментальні засоби проведення рефакторингу. Літ.: [1,6]	Лабораторна робота № 3. Частина 1. Рефакторінг та оптимізація програмного коду. Літ.: [1,6]	Практична робота №5. Операції реінжинірингу та рефакторингу архітектури. Літ.: [1,4,10]	Опрацювання конспекту лекцій Т4; опрацювання навчальних посібників	10	[1; 5]
6	Реінжиніринг та рефакторінг баз даних. Категорії операцій реінжинірингу та рефакторингу бази даних. Процеси реінжинірингу та рефакторингу бази даних Літ.: [2,5,10]	Лабораторна робота № 3. Частина 2. Рефакторінг та оптимізація програмного коду. Літ.: [1,6]	Практична робота №6. Методи та методики оптимізації коду. Літ.: [1-4]	Опрацювання конспекту лекцій Т5; підготовка до лабораторного заняття	10	[1; 5]
7	Операції реінжинірингу та рефакторингу бази даних. Операції реінжинірингу та рефакторингу структури бази даних [2,5,110]	Лабораторна робота №4. Частина 1. Реінженерія та оптимізація БД. Оптимізація взаємодії програмної системи і СКБД. Літ.: [2,5]	Практична робота №7. Операції рефакторингу якості даних. Літ.: [1,2,5]	Опрацювання конспекту лекцій Т6; опрацювання навчальних посібників	10	[1, 2]
8	Аналіз критеріїв ефективності програмних систем. Методи і моделі визначення	Лабораторна робота №4. Частина 2. Реінженерія та оптимізація	Практична робота №8. Операції рефакторингу бази даних	Опрацювання конспекту лекцій Т7; підготовка до лабораторного заняття	10	[1, 2, 3]

	ефективності використання програмних систем. Нечітко-множинні підходи й критерії оцінки програмних систем.[1-6,9]	БД. Оптимізація взаємодії програмної системи і СКБД. Літ.: [2,5]	Літ.: [2,5]			
9	Оптимізаційні моделі розробки програмного забезпечення з показниками величини проекту. Оптимізація на основі моделі заснованої на кількості помилок . Функція інтенсивності виявлення несправностей . Модель з індексом величини проекту. Процедури стохастичної оптимізації .Літ.: [1-6, 9]			Опрацювання конспекту лекцій Т8-Т9	10	[3; 4].

Примітка: *- Лекції проводяться раз в два тижні по дві години, лабораторного заняття проводяться раз у два тижня по дві години по чисельнику чи знаменнику, відповідно до розкладу занять.

Політика дисципліни

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу <https://khmnu.edu.ua/wp-content/uploads/normativni-dokumenty/polozhennya/pro-organizacziyu-osvitnogo-proczesu.pdf>, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції і лабораторного заняття згідно з розкладом, не запізнюватися на заняття, домашні завдання виконувати відповідно до графіка. Пропущене лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. Термін виконання лабораторної роботи та кейсів на лабораторному занятті вважається своєчасним, якщо здобувач виконав їх на наступному занятті після отримання завдання. До лабораторних занять студент має підготуватися за відповідною темою і проявляти активність. Набутті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (вебсайт Університету (<https://khmnu.edu.ua/>): розділ «Нормативні документи», рубрика – «Положення», сторінка – «Положення про організацію освітньої діяльності»).

Здобувачі вищої освіти при вивченні дисципліни можуть користуватись власними пристроями (ноутбуками, планшетами, смартфонами). Власними пристроями можна користуватися як для роботи в системі Moodle, так і для доступу до зовнішніх інформаційних ресурсів, які необхідні для виконання практичних занять.

Під час роботи над завданнями лабораторної роботи студент має дотримуватися політики доброчесності, яка викладена у Кодексі академічної доброчесності учасників освітнього процесу Хмельницького національного університету (<https://vzia.khmnu.edu.ua/wp-content/uploads/sites/5/kodeks-akademichnoyi-dobrochesnosti.pdf>). У разі виявлення плагіату здобувач вищої освіти отримує незадовільну оцінку і має виконати нове завдання за темою.

Критерії оцінювання результатів навчання

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з урахуванням коефіцієнта вагомості і встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу. Вагові коефіцієнти залежать від структури дисципліни і важливості окремих видів її робіт. При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість набуття теоретичних знань і практичних навичок перевіряється шляхом вирішення практичних завдань та кейсів. Оцінка, яка виставляється за практичне заняття, складається з таких елементів: знання теоретичного матеріалу з теми; вміння студента обґрунтувати прийняті рішення та розв'язувати практичні завдання.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота	Самостійна, індивідуальна робота	Семестровий контроль (залік)
усне опитування, розв'язування практичних завдань (обов'язковий мінімум оцінок – 4)	Тестовий контроль:	Іспит

Т 1-9	Т 1-9	
ВК: 0,4	0,2	0,4

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт.

Оцінювання тестових завдань

Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом (може бути інший варіант). Максимальна сума балів, яку може набрати студент, складає 20.

Оцінювання здійснюється за чотирибальною шкалою. Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Відповідність набраних балів за тестове завдання вітчизняній оцінці

Сума балів за тестове завдання	1–10	11–12	13–15	17-20
Оцінка	2	3	4	5

На тестування відводиться 20 хвилин (для закритої форми тестів – по одній хвилині на кожне завдання). Правильні відповіді студент записує у талоні відповідей. При цьому усі граfi для відповідей мають бути заповнені цифрами, що відповідають правильним, на погляд студента, відповідям. Через 20 хвилин студенти здають викладачу завдання з талонами відповідей. Викладач на наступному занятті оголошує результати тестування. Тестування студент може також пройти і в он-лайн режимі в модульному середовищі MOODLE.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у таблиці.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Інституційна шкала балів	Інституційна оцінка	Критерії оцінювання
A	4,75-5,00	5	Зараховано Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок. Добре – повне знання навчального матеріалу з кількома незначними помилками. Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками. Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією. Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
B	4,25-4,74	4	
C	3,75-4,24	4	
D	3,25-3,74	3	
E	3,00-3,24	3	
FX	2,00-2,99	2	Незараховано Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни.
F	0,00-1,99	2	

Підсумкова семестрова оцінка виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться у межах від 3,00 до 5,00 балів. При цьому за інституційною шкалою ставиться оцінка «задовільно, добре, відмінно», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів відповідно до таблиці співвідношення.

Питання для самоконтролю результатів навчання

1. Ознаки продуктивності програмного забезпечення.
2. Надійність програмних систем.
3. Програмні та організаційні заходи для визначення продуктивності ПЗ.
4. Процеси розробки, їх закони, правила та умови проектування.
5. Реінжиніринг програмних систем
6. Етапи та процеси реінжинірингу програмного забезпечення.
7. Специфікації завдань.
8. Типи архітектур. Оптимізація робіт на етапі проектування програмної системи
9. Типи кодування.
10. Мови, типи платформ, кросплатформеність.
11. Оптимальне конструювання модулів програмних систем
12. Продуктивність у суміжних областях ІТ.

13. Зворотня розробка програмного забезпечення
14. Рефакторинг програмних систем
15. Оптимізація постійних виразів.
16. Організація та оптимізація даних.
17. Організація та оптимізація програмного коду.
18. Організація та оптимізація алгоритмів.
19. Частота оновлення.
20. Частота фіксації помилок.
21. Заміна змінних константними значень ("розмноження" констант).
22. Вирахування значень змінних на стадії компіляції (константа "згорання").
23. Вирахування значень функцій на стадіях компіляції ("згортка" функцій).
24. Оптимізація алгебраїчних виразів.
25. Видалення невикористовуваних змінних.
26. Видалення копій змінних.
27. Видалення невикористовуваних присвоювань.
28. Видалення зайвих виразів.
29. Видалення зайвих викликів функцій.
30. Виконання алгебраїчних спрощень.
31. Основні методи оптимізації.
32. Оптимізація підвиразів.
33. Оптимізація арифметичних операцій.
34. Оптимізація розгалужень.
35. Заміна переходів арифметичних операцій.
36. Видалення зайвих умов.
37. Видалення завідомо хибних умов.
38. Оптимізація switch.
39. Балансування логічного дерева.
40. Створення таблиць переходів.
41. Оптимізація циклів.
42. Оптимізація викликів функцій.
43. Оптимізація передачі аргументів.
44. Оптимізація функцій. Оптимізація розподілу змінних.
45. Оптимізація ініціалізації строк.
46. Оптимізація "мертвого" коду. Оптимізація константних умов.
47. Реінжиніринг та оптимізація даних у СКБД.
48. Операції реінжинірингу та рефакторингу структури бази даних
49. Старіння процедурного кешу.
50. Неоптимальні індекси.
51. Неоптимальні статистики.
52. Оптимальне конструювання модулів програмних систем.
53. Критерії оцінки якості машинної оптимізації.
54. Методики оцінки якості машинної оптимізації.

Рекомендована література

Основна

1. Великодній С. С., Тимофеева О. С. Реінжиніринг програмного забезпечення інформаційних систем: монографія / С. С. Великодній, О. С. Тимофеева. – Одеса: Видавничий дім «Гельветика», 2020. – 160 с. URL: http://eprints.library.odku.edu.ua/id/eprint/8232/1/VelykodniySS_Reingining_programnogo_zabezpechenia_2020.pdf
2. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фабула», 2019. – 448 с.
3. Fowler Martin. Refactoring: Improving the Design of Existing Code (2nd Edition). Martin Fowler. Addison-Wesley Professional. 2019. – 424 p.
4. Доценко С. І. Організація та системи керування базами даних: Навч. посібник. – Харків: УкрДУЗТ, 2023. – 117 с.
5. Форкун Ю., Бойко В. Удосконалення методу матеріалізованих представлень у реінжинірингу бази даних. Вимірювальна та обчислювальна техніка в технологічних процесах 2022, № 3 с. 87-91
6. Форкун Ю., Мартинюк В., Праворська Н., Луцицький О. (2023). Метрика диференційованої цикломатичної складності аналізу програмного коду з використанням систем керування базами даних. Measuring and computing devices in technological processes, (3, 2023), с. 100–105. <https://doi.org/10.31891/2219-9365-2023-75-11>

Додаткова

1. Martin Fowler. Refactoring: Improving the Design of Existing Code (2nd Edition). Addison-Wesley Professional, 2019. – 424p. ISBN 978-0134757599
2. Scott Ambler. Refactoring Databases: Evolutionary Database Design. Addison-Wesley Signature Series,

2017. – 384 p. ISBN 978-0-321-77451-4

3. Con Mike User Stories Applied for Agile Software Development. Addison-Wesley Professional, 2021. – 268 p. ISBN 0-321-20568-5

4. Len Bass, Paul Clements, Rick Kazman. Architektura oprogramowania w praktyce. Wydanie II. Helion. 2021, - 464 s. ISBN 978-83-246-3302-9

2. SQL and NoSQL Databases. Second Edition. Michael Kaufmann, Andreas Meier/ Springer Cham, 2023. – 254 p. <https://doi.org/10.1007/978-3-031-27908-9>

3. SQL and NoSQL interview questions. Vishwanathan Narayanan. BPB PUBLICATIONS, [S.l.], 2023. – 178p.

4. Alan Beaulieu Learning SQL: Generate, Manipulate, and Retrieve Data/Beaulieu Alan - O'Reilly Media 2020. - 377 pages.

5. Carlos Coronel. Database Systems: Design, Implementation, & Management (MindTap Course List) / Coronel Carlos, Morris Steven – Boston, Massachusetts: Cengage Learning – 14th edition, 2022 – 816 p.

6. Ying Bai. SQL Server Database Programming with Visual Basic.NET/Ying Bai – Wiley, 2020– 688p.

7. A workload-driven method for designing aggregate-oriented NoSQL databases. Liu Chen, Ali Davoudian, Mengchi Liu. Data & Knowledge Engineering Volume 142, November 2022, 102089

8. Designing NoSQL databases based on multiple requirement views. Roy- Hubara, Arnon Sturm, Peretz Shoval. Data & Knowledge Engineering Volume 145, May 2023, 102149

9. Security&privacy issues and challenges in NoSQL databases. Sabrina Sicari, Alessandra Rizzardi, Alberto Coen-Porisini. Computer Networks Volume 206, 7 April 2022, 108828

10. A unified metamodel for NoSQL and relational databases. Carlos J.Fernández Candel, Diego Sevilla Ruiz, Jesús J. García-Molina. Information Systems Volume 104, February 2022, 101898

Розробник



к.т.н., доцент Юрій ФОРКУН

Погоджено

Гарант

ОПП



к.т.н., доцент Оксана ЯШИНА

Завідувач кафедри

Інженерії програмного забезпечення



д.ф.-м.н., професор Леонід БЕДРАТЮК