



Міжнародний гуманітарний університет
Факультет кібербезпеки, програмної інженерії
та комп'ютерних наук
Кафедра інформаційних технологій

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
РЕІНЖИНІРІНГ ТА ОПТИМІЗАЦІЯ ПРОГРАМНИХ СИСТЕМ

Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
Назва освітньої програми	Інженерія програмного забезпечення
Рівень вищої освіти	другий (магістерський) рівень

Розробники і викладачі <i>(азначаються розробники і викладачі, які викладають дисципліну - посада, наук. ступінь, вчене звання, П.І.Б.)</i>	Контактний тел.	E-mail
професор кафедри інформаційних технологій, д.т.н., професор Приходько Сергій Борисович	+380993876131	sergiy.prykhodko@nuos.edu.ua

1. АНОТАЦІЯ ДО КУРСУ

Вивчення дисципліни «**Реінжинірінг та оптимізація програмних систем**» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач наукової діяльності, пов'язаної із роботою з існуючим програмним забезпеченням, його вдосконаленням та масштабуванням для задоволення сучасних потреб супроводження такого програмного забезпечення.

Предметом дисципліни «**Реінжинірінг та оптимізація програмних систем**» є математичне та алгоритмічне забезпечення процесів переробки вихідного коду.

Метою навчальної дисципліни є формування у здобувачів вищої освіти таких компетентностей:

- Здатність розробляти якісне та надійне програмне забезпечення складних програмних комплексів та систем на основі новітніх технологій та стандартів розроблення програмного забезпечення.
- Здатність проводити експериментальні дослідження з оцінювання ефективності та безпечності програмного забезпечення.
- Здатність самостійно виконувати науково-дослідну діяльність в інженерії програмного забезпечення із застосуванням сучасних концепцій, методів, та технологій.
- Здатність критично переосмислювати наявні технології інженерії програмного забезпечення та відстежувати тенденції їх розвитку.

Отримані в результаті засвоєння дисципліни «Реінжинірінг та оптимізація програмних систем» теоретичні знання та практичні уміння є корисними для проведення наукових досліджень за темою дисертації.

2. ОЧІКУВАНІ КОМПЕТЕНТНОСТІ, ЯКІ ПЛАНУЄТЬСЯ СФОРМУВАТИ ТА ДОСЯГНЕННЯ ПРОГРАМНИХ РЕЗУЛЬТАТІВ

Інтегральна компетентність

ІК01. Здатність особи розв'язувати складні задачі і проблеми у певній галузі професійної діяльності або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов і вимог.

Загальні компетентності (ЗК)

ЗК04. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами інших галузей знань/видів економічної діяльності).

Спеціальні (фахові) компетентності

СК03. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.

СК04. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.

СК10 Здатність розробляти моделі та засоби інтелектуальної обробки даних в розподілених системах, вдосконалювати програмні системи та здійснювати оптимізацію програмного забезпечення з урахуванням вимог до їх надійності.

Навчальна дисципліна «*Реінжинірінг та оптимізація програмних систем*» забезпечує досягнення програмних результатів навчання (РН), передбачених освітньою програмою:

РН06 Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

РН08 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.

РН10 Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.

РН11 Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

РН12 Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики.

РН13 Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

РН15 Здійснювати реінжинірінг програмного забезпечення відповідно до вимог замовника.

PH16 Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

PH 18 Розробляти моделі та засоби обробки даних в розподілених системах та здійснювати оптимізацію програмного забезпечення з урахуванням вимог до надійності.

3. ОБСЯГ ТА ОЗНАКИ КУРСУ

Загалом		Вид заняття (денне відділення / заочне відділення)			Ознаки курсу		
ЄКТС	годин	Лекційні заняття	Практичні заняття	Самостійна робота	Курс, (рік навчання)	Семестр	Обов'язкова / вибіркова
5	150	42/6	28/6	80/138	1	1	Обов'язкова

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	денна форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекц.	прак	сам. роб.		лекц.	прак	сам. роб.
Тема 1. Методологія реінжинірингу.	36	10	6	20	42	2	-	40
Тема 2. Дослідження й розробка шаблонів реінжинірингу.	34	8	6	20	34	2	2	30
Тема 3. Технології та засоби реінжинірингу.	40	12	8	20	40	-	2	38
Тема 4. Оптимізація програмних систем.	40	12	8	20	34	2	2	30
Усього годин	150	42	28	80	150	6	6	138
ПІДСУМКОВИЙ КОНТРОЛЬ - ЕКЗАМЕН								

5. ТЕХНІЧНЕ Й ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ / ОБЛАДНАННЯ

Здобувачі отримують теми та питання курсу, основну і додаткову літературу, рекомендації, завдання та оцінки за їх виконання як традиційним шляхом, так і з використанням університетської платформи онлайн навчання на базі Moodle та на базі Google Клас. Окрім того, практичні навички у пошуку та аналізі інформації за курсом, з оформлення індивідуальних завдань, тощо, студенти отримують, користуючись університетськими комп'ютерними класами та бібліотекою.

6. САМОСТІЙНА РОБОТА

До самостійної роботи студентів щодо вивчення дисципліни «Реінжинірінг та оптимізація програмних систем» включаються:

1. Знайомство з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.
3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка та виконання індивідуальних завдань, доповідей, рефератів тощо.
7. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 1. Методологія реінжинірінгу. Поняття рефакторінгу, чистого та брудного коду, запаху коду. Актуальні завдання у проблематиці реінжинірінгу. Оцінювання запаху коду. Уніфікація і стандартизація поняття «реінжинірінг ІС». Розробка цілісних методологій реінжинірінгу ІС. Рівень методології й рівень технології реінжинірінгу ІС. Огляд різних методик модернізації успадкованих систем: заміщення, модернізація, еволюція. Моделі модернізації систем. Каркасна модель. Технологія KADS для проектування ІС на основі шаблонів. Реінжинірінг ПЗ у проектуванні Веб-додатків.	20	40
2	Тема 2. Дослідження й розробка шаблонів реінжинірінгу. Патерни (шаблони), переваги та недоліки їх використання. Історія створення патернів. Класифікація патернів. Породжуючі патерни. Структурні патерни. Поведінкові патерни. Патерни проектування об'єктів інформаційних систем. Архітектурні і системні патерни. Патерни інтеграції інформаційних систем.	20	30
3	Тема 3. Технології та засоби реінжинірінгу. Засоби реінжинірінгу в мовах програмування. Темплейти і суперкласи в С++, С#. Призначення і функції суперкласів. Шаблони класів. Реінжинірінг ПЗ на основі методу реплікації баз даних з використанням об'єктно-орієнтованого підходу. Уніфіковані програмні засоби для задач реінжинірінгу. Макроси, бібліотеки, репозитарії, сховища даних, оболонки. Технологія NET Framework. Характеристики і функції. Тенденції і напрямки розвитку технологій реінжинірінгу. Можливості проблемно-орієнтованих технологій. Хмарні технології для реінжинірінгу. Задачі реінжинірінгу бізнес-процесів. Моделі бізнес-процесів. Технології реінжинірінгу бізнес-процесів VI та VAM. Технології функціонального моделювання бізнес-процесів - IDEF0, IDEF3. Використання діаграм DFD для опису потоків даних.	20	38
4	Тема 4. Оптимізація програмних систем. Оптимізація як бажана ціль у розробці ПЗ. Пошук компромісу між оптимізацією та стабільністю, зручністю обслуговування та переносимістю ПЗ. Загальна постановка задачі оптимізації коду. Продуктивність програми.	20	30

Архітектура коду та її вплив на продуктивність програми. Підвищення продуктивності програми за рахунок застосування мов низького рівня. Приклади хороших прийомів оптимізації коду. Усунення розгалуження як важлива проблема для сучасних глибоко конвеєрних архітектур процесорів. Оптимізація при виконанні циклу. Оптимізація при використанні масивів.		
Всього	80	138

7. ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Робоча програма навчальної дисципліни передбачає наступні види та методи контролю:

Види контролю	Складові оцінювання
поточний контроль , який здійснюється у ході: проведення практичних занять, виконання індивідуального завдання; проведення консультацій та відпрацювань.	50%
підсумковий контроль , який здійснюється у ході проведення іспиту.	50%

Методи діагностики знань (контролю)	фронтальне опитування; наукова доповідь, рефера, усне повідомлення, індивідуальне опитування, практичних завдань, іспит.
--	--

8. ОЦІНЮВАННЯ ПОТОЧНОЇ, САМОСТІЙНОЇ ТА ІНДИВІДУАЛЬНОЇ РОБОТИ СТУДЕНТІВ З ПІДСУМКОВИМ КОНТРОЛЕМ У ФОРМІ ЕКЗАМЕНУ/ ЗАЛІКУ

Денна форма навчання			
<i>Поточний контроль</i>			
Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на семінарських (практичних) заняттях			
1.1. Підготовка до практичних занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних занять	25
Виконання завдань для самостійного опрацювання			
1.2. Підготовка програмного матеріалу (тем, питань), що виносяться на самостійне вивчення	Відповідно до робочої програми та розкладу занять	Розгляд відповідного матеріалу під час аудиторних занять або ІКР ¹ , перевірка конспектів навчальних текстів тощо	10
Виконання індивідуальних завдань (науково-дослідна робота студента)			

¹ Індивідуально-консультаційна робота викладача зі студентами

1.3. Підготовка реферату за заданою тематикою	Відповідно до робочої програми та розкладу занять	Обговорення (захист) матеріалів реферату	10
1.4. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять або ІКР, наукових конференцій та круглих столів.	5
Разом балів за поточний контроль			50
<i>Підсумковий контроль екзамен</i>			50
Всього балів			100

9. КРИТЕРІЇ ПІДСУМКОВОЇ ОЦІНКИ ЗНАНЬ СТУДЕНТІВ (для іспиту / заліку)

Рівень знань оцінюється:

- «відмінно» / «зараховано» А - від 90 до 100 балів. Студент виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

- «добре» / «зараховано» В - від 82 до 89 балів. Студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

- «добре» / «зараховано» С - від 74 до 81 балів. Студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;

- «задовільно» / «зараховано» D - від 64 до 73 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, рефератів (есе);

- «задовільно» / «зараховано» E - від 60 до 63 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

- «незадовільно з можливістю повторного складання» / «не зараховано» FX – від 35 до 59 балів. Студент володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.

- «незадовільно з обов'язковим повторним вивченням дисципліни» / «не зараховано» F – від 0 до 34 балів. Студент не володіє навчальним матеріалом.

Таблиця відповідності результатів контролю знань за різними шкалами

100-бальною шкалою	Шкала за ECTS	За національною шкалою	
		екзамен	залік
90-100 (10-12)	A	Відмінно	зараховано
82-89 (8-9)	B	Добре	
74-81(6-7)	C		
64-73 (5)	D	Задовільно	
60-63 (4)	E		
35-59 (3)	Fx	незадовільно	не зараховано
1-34 (2)	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фабула», 2019. – 448 с.
2. Програмна інженерія: підруч. / К.М. Лавріщева; Ін-т програм. систем НАН України. — К.: Академперіодика, 2008. — 320 с.
3. Fowler, M. Refactoring: Improving the Design of Existing Code Edition Unstated / Martin Fowler. – Addison-Wesley Professional, 2018. – 448 р.
4. Michael McCool, James Reinders, Arch Robison. Structured Parallel Programming: Patterns for Efficient Computation. 2012. – 432 p.

Допоміжна

1. Підхід до оптимізації програмного забезпечення для аналізу великих даних / А.М. Лавренюк, С.І. Лавренюк, П.Г. Тульчинський // Комп'ютерна математика. — 2017. — № 1. — С. 121-125.
2. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О. Б. Данченко. Київ: Університет економіки та права «КРОК», 2017. – 238 с.
3. Michael Feathers. Working Effectively with Legacy Code. – Person, 2004. – 464 p.
4. G. Szóke, C. Nagy, R. Ferenc and T. Gyimóthy, "Designing and Developing Automated Refactoring Transformations: An Experience Report," *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Osaka, Japan, 2016, pp. 693-697, doi: 10.1109/SANER.2016.17.
5. F. Coelho, T. Massoni and E. L.G. Alves, "Refactoring-Aware Code Review: A Systematic Mapping Study," *2019 IEEE/ACM 3rd International Workshop on Refactoring (IWor)*, Montreal, QC, Canada, 2019, pp. 63-66, doi: 10.1109/IWoR.2019.00019.
6. N. Yoshida, S. Numata, E. Choiz and K. Inoue, "Proactive Clone Recommendation System for Extract Method Refactoring," *2019 IEEE/ACM 3rd International Workshop on Refactoring (IWor)*, Montreal, QC, Canada, 2019, pp. 67-70, doi: 10.1109/IWoR.2019.00020.

7. R. S. Bashir, S. P. Lee, C. C. Yung, K. A. Alam and R. W. Ahmad, "A Methodology for Impact Evaluation of Refactoring on External Quality Attributes of a Software Design," *2017 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, 2017, pp. 183-188, doi: 10.1109/FIT.2017.00040.
8. N. A. Nagy and R. Abdalkareem, "On the Co-Occurrence of Refactoring of Test and Source Code," *2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, Pittsburgh, PA, USA, 2022, pp. 122-126, doi: 10.1145/3524842.3528529.
9. M. Mohan, D. Greer and P. McMullan, "Maximizing Refactoring Coverage in an Automated Maintenance Approach Using Multi-Objective Optimization," *2019 IEEE/ACM 3rd International Workshop on Refactoring (IWoR)*, Montreal, QC, Canada, 2019, pp. 31-38, doi: 10.1109/IWoR.2019.00014.
10. A. Brito, A. Hora and M. T. Valente, "Refactoring Graphs: Assessing Refactoring over Time," *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, London, ON, Canada, 2020, pp. 367-377, doi: 10.1109/SANER48275.2020.9054864.
11. M. Agnihotri and A. Chug, "Understanding Refactoring Tactics and their Effect on Software Quality," *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2022, pp. 41-46, doi: 10.1109/Confluence52989.2022.9734141.
12. J. Pérez, "Refactoring Planning for Design Smell Correction: Summary, Opportunities and Lessons Learned," *2013 IEEE International Conference on Software Maintenance*, Eindhoven, Netherlands, 2013, pp. 572-577, doi: 10.1109/ICSM.2013.98.
13. A. Pizzini, "Behavior-based test smells refactoring : Toward an automatic approach to refactoring Eager Test and Lazy Test smells," *2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, Pittsburgh, PA, USA, 2022, pp. 261-263, doi: 10.1145/3510454.3517059.
14. S. Jindal and G. Khurana, "The statistical analysis of source-code to determine the refactoring opportunities factor (ROF) using a machine learning algorithm," *Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2013)*, Bangalore, 2013, pp. 396-403, doi: 10.1049/cp.2013.2244.
15. Prykhodko S. A Statistical Evaluation of the Depth of Inheritance Tree Metric for Open-Source Applications Developed in Java / S. Prykhodko, N. Prykhodko, T. Smykodub // *Foundations of Computing and Decision Sciences*. – 2021. – Vol. 46. – No. 2. – P. 159-172. – ISSN 0867-6356, e-ISSN 2300-3405. DOI: <https://doi.org/10.2478/fcds-2021-0011>
16. Prykhodko S. A Joint Statistical Estimation of the RFC and CBO Metrics for Open-Source Applications Developed in Java / S. Prykhodko, N. Prykhodko, T. Smykodub // *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, 2022. – P. 442-445. DOI: <https://doi.org/10.1109/CSIT56902.2022.10000457>
17. Prykhodko S.B. A statistical estimation of the coupling between object metric for open-source apps developed in Java / S. B. Prykhodko, K. S. Prykhodko, T. G. Smykodub // *Herald of Advanced Information Technology*. – 2022. – No. 3 (5). – P. 175-184. DOI: <https://doi.org/10.15276/hait.05.2022.13>

Інформаційні ресурси

1. Національна бібліотека України імені В. І. Вернадського [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: НБУВ, 2013-2015. – Режим доступу: www.nbuv.gov.ua
2. Електронний каталог Національної парламентської бібліотеки України [Електронний ресурс]: [політемат. база даних містить відом. про вітчизн. та зарубіж. кн., брош., що надходять у фонд НПБ України]. – Електронні дані (803 438 записів). – Київ: Нац. парлам. б-ка України, 2002-2015. – Режим доступу: catalogue.nplu.org
3. Український інститут інтелектуальної власності [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: УІІВ, 2017. – Режим

доступу: <http://www.uipv.org>

4. Paul Hsieh. Programming Optimization. – [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <http://www.azillionmonkeys.com/qed/optimize.html>
5. An introduction to the owl web ontology language. Режим доступа: <http://www.cse.lehigh.edu/~heflin/IntroToOWL.pdf>
6. Idiomatic Ways to Refactor Your Python Code / Yong Cui. – [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://towardsdatascience.com/10-idiotic-ways-to-refactor-your-python-code-cbb05bb0c820>