МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Затверджено
На засіданні кафедри РКС
факультету електроніки та комп'ютерних технологій
Львівського національного університету імені Івана Франка
(протокол № ____ від 2022 р.)

Завідувач кафедри
_____ Ігор ОЛЕНИЧ

Силабус з навчальної дисципліни «Аналіз та рефакторинг коду, моделювання та проектування ПЗ», що викладається в межах ОПП " Інженерія програмного забезпечення " першого (бакалаврського) рівня вищої освіти для здобувачів з спеціальності 121 – Інженерія програмного забезпечення

Львів 2022

Назва дисципліни	Аналіз та рефакторинг коду, моделювання та проектування ПЗ
Адреса викладання дисципліни	м. Львів, вул. Драгоманова, 50
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіоелектронних і комп'ютерних систем
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 121 Інформаційні системи та технології
Викладачі дисципліни	Монастирський Л.С., докт., фіз.-мат. наук, професор Гура В.Т., асистент
Контактна інформація викладачів	lyubomyr.monastyrskyy@lnu.edu.ua volodymyr.gura@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/hura-volodymyr-tarasovych
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams та Telegram (у текстовому режимі). Для погодження часу онлайн консультацій слід писати на електронну пошту викладача, або в особистий Telegram.
Сторінка дисципліни	https://electronics.lnu.edu.ua/course/qa
Інформація про дисципліну	Дисципліна «Аналіз та рефакторинг коду, моделювання та проектування ПЗ» є дисципліною з спеціальності 126 Інформаційні системи та технології для освітньої програми «Інформаційні системи та технології», яка викладається в 6 семестрі в обсязі 6.0 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб надати студентам необхідні знання на практичні навички для роботи зі складними процесами моделювання та аналізу якості програмного забезпечення(ПЗ). Такі знання та навички слугують базисом для того, щоб вміти реалізовувати та застосовувати типові алгоритми сценаріїв та відповідні їм структури простого коду для вирішення задач стабільності та якості програмного продукту. Тому у дисципліні представлено як теоретичні та практичні основи моделей тестування та конкретні реалізації бібліотеки, володіння якими є необхідним для створення прототипів та розгортання моделей простих фреймворків для неперервного процесу інтегрування нового коду. Курс інтегрований з програмною базами (на основі .Net і Specflow).
Мета та цілі дисципліни	Метою вивчення вибіркової дисципліни 'Аналіз та рефакторинг коду, моделювання та проектування ПЗ' є одержання студентами знань про види ПЗ та сучасні підходи до програмної розробки таких систем з використанням мови програмування .Net; опанування навичок із принципам SOLID;набуття студентами практичних умінь аналізу та рефакторинк до ПЗ. Програма передбачає вивчення патерніп проектування та застосування їх до власних проєктів.
Література для вивчення дисципліни	Основна література: 1. Scott W. Ambler and Pramod J. Sadalage. Refactoring Databases. Addison-Wesley, 2006. 2. Jay Fields, Shane Harvie, and Martin Fowler. Refactoring Ruby Edition. Addison-Wesley, 2009 3. William C. Wake. Refactoring Workbook. Addison-Wesley, 2003. ISBN: 0321109295 4. Michael Feathers. Working Effectively with Legacy Code. Prentice Hall, 2004.

	5. Joshua Kerievsky. Refactoring to Patterns. Addison-Wesley, 2004. 6. B.Wagner Effective C#:5 Specific Ways to Improve Your C#, 2015,224p. 7. Joseph Albahari, Ben Albahari C# 7.0 in a Nutshell: The Definitive Reference,2018, 1070 p. 8. Head First Design Patterns: A Brain-Friendly Guide: Building Extensible and Maintainable Object-Oriented Software Paperback – 5 Jan. 2021 9. Refactoring: Improving the Design of Existing Code (Addison-Wesley Signature Series (Fowler)) Hardcover – Illustrated, 2 Jan. 2019 10. Domain-Driven Design: Tackling Complexity in the Heart of Software Hardcover – Illustrated, 11 Sept. 2013 11. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions (Addison-Wesley Signature Series (Fowler)) Hardcover – 23 Oct. 2013.
Обсяг курсу	64 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт та 116 годин самостійної роботи.
Очікувані результати навчання	Після завершення цього курсу студент буде: - Знати сучасні тенденції розвитку інформаційних систем і технологій, демонструвати практичні навички програмування та використання прикладних і спеціалізованих комп’ютерних систем і комунікаційних технологій у професійній діяльності; процес створення чітко структурованих програм на мові C# під потреби ПЗ; основи інтегрованих процесів; механізм розгортання написаних CI/CD рішень, а також інший теоретичний і практичний матеріал згідно програми курсу. - Вміти застосовувати патерни для створення програмного забезпечення; створювати та використовувати сучасні фреймворк архітектури для потреб моделювання даних; демонструвати навички аназізу засобів інформаційних технологій під час проектування із розроблення IT рішень для розвитку та підтримки та стабільності. Після вивчення даного курсу здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання: ЗК1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК2. Здатність застосовувати знання у практичних ситуаціях. ЗК4. Здатність спілкуватися іноземною мовою як усно, так і письмово. ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу. ФК19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення. ФК28. Володіння методами сучасних веб-технологій, хмарних технологій, великих даних та засобами розробки веб-застосувань. ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи сучасні парадигми програмування.

	ПРН2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності. 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. ПРН9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення. ПРН10. Проводити передпроєктне обстеження предметної області, системний аналіз об'єкта проектування. ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення. 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. ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення. ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.
Ключові слова	.Net, NUnit, BDD, CI/CD, SQL, SOLID.
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін « «Основи програмування».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
Необхідне обладнання	Мультимедіа, платформа Microsoft Teams, комп’ютерне та апаратне програмне забезпечення.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням: • лабораторні роботи: 60% семестрової оцінки; максимальна кількість балів 60. • контрольні заміри (2 модулі): 40% семестрової оцінки; максимальна кількість балів 40. Загалом упродовж семестру 100 балів. **Контрольні заміри проводяться у формі тестових завдань. Академічна доброчесність**: Очікується, що лабораторні та контрольні роботи студентів будуть їх оригінальними дослідженнями чи

	міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахуванння викладачем, незалежно від масштабів плагіату чи обману. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом. **Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. **Політика виставлення балів.** Враховуються бали набрані на поточному тестуванні та самостійній роботі. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнень на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання. Жодні форми порушення академічної доброчесності не толеруються.
Питання до контрольних робіт	Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт розміщені на веб-сторінці.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	Refactoring: Improving the Design of Existing Code	Лекція	1, 2		2 тиж. семестру
2	What is Refactoring	Лекція	1, 2, 9		3 тиж. семестру
3	Code Smells	Лекція	1, 4, 9	Пошук недоліків у вланних проєктах, 4 год.	4 тиж. семестру
4	Technical debt	Лекція	1, 9	Реалізація технічної заборгованості, 2 год.	5 тиж. семестру
5	Refactoring Techniques	Лекція	1, 4, 9	Робота з рефакторінгом, 3	6 тиж. семестру

				год.	
6	Refactoring Techniques: Organization Data	Лекція	1,9	Реалізація декількох методів рефакторінгу даних, 4 год.	7 тиж. семестру
7	How to refactor	Лекція	2		8 тиж. семестру
8	Analyze Refactoring in projects	Лекція	2,3	Аналіз проєтку, 2 год.	9 тиж. семестру
9	Design patterns	Лекція	8, 10	Робота з патернами, 4 год.	10 тиж. семестру
10	Benefits of patterns, Classification	Лекція	7, 8		12 тиж. семестру
11	Design patterns: Creational patterns	Лекція	8, 11	Робота над моделлю та архітектурою проєтку	13 тиж. семестру
12	Design patterns: Structural patterns	Лекція	8, 10	Класифікація підходів до патернів, 4 год.	14 тиж. семестру
13	Design patterns: Behavioral patterns	Лекція	8, 10	Створення проекту, 2 год.	15 тиж. семестру
14, 15	Design patterns: implementation	Лекція	8, 11	Аналіз чужого коду, 4 год.	16 тиж. семестру
16	Підсумкове заняття. Рефекторінг на проєктах. Подальші кроки самовдосконалення.	Лекція	8		