

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Кафедра (циклова комісія) _____ радіоелектронних і комп'ютерних систем

“ЗАТВЕРДЖУЮ”

Декан факультету _____

доц. Юрій ФУРГАЛА

“ ”

2022 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**АНАЛІЗ ТА РЕФАКТОРИНГ КОДУ,
МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПЗ**

(шифр і назва навчальної дисципліни)

спеціальність _____ 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

спеціалізація _____

(назва спеціалізації)

факультет _____ електроніки та комп'ютерних технологій

(назва інституту, факультету, відділення)

Робоча програма “Аналіз та рефакторинг коду, моделювання та проектування ПЗ” для студентів

(назва навчальної дисципліни)

галузі знань “12 – Інформаційні технології”

за спеціальністю “121 Інженерія програмного забезпечення”

Розробники: Гура Володимир (асистент)

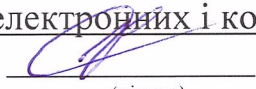
(вказати авторів, їхні посади, наукові ступені та вчені звання)

Робочу програму схвалено на засіданні кафедри (циклової комісії) _____

радіоелектронних і комп’ютерних систем

Протокол від “ 31 ” 08 2022 року № 1/23

Завідувач кафедри радіоелектронних і комп’ютерних систем

 (Оленич І.Б.)
(підпис) (прізвище та ініціали)

Ухвалено Вченою радою факультету електроніки та комп’ютерних технологій

Протокол від “ 31 ” 08 2022 року № 28/22

© Гура В., 2022 рік

© ЛНУ ім. І. Франка, 2022 рік

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів 4	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Нормативна (за вибором)	
Модулів – <i>немає</i>	Спеціальність: <u>121 Інженерія програмного забезпечення</u>	Рік підготовки	
Змістових модулів – 2		3-й	
Індивідуальне науково-дослідне завдання <u>немає</u> (назва)		Семестр	
Загальна кількість годин – 64		6-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 4	Освітній ступінь <u>бакалавр</u>	Лекції	
		32 год.	
		Практичні, семінарські	
		<i>немає</i>	
		Лабораторні	
		32 год.	
		Самостійна робота	
		116 год.	
		Індивідуальні завдання:	
<i>немає</i>			
Вид контролю:			
<i>екзамен</i>			

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить

для денної форми навчання – 1

для заочної форми навчання – немає

2.

Мета та завдання навчальної дисципліни

Мета: одержання студентами знань про види ПЗ та сучасні підходи до програмної розробки таких систем з використанням мови програмування .Net

Цілі: опанування навичок із принципам SOLID; набуття студентами практичних умінь аналізу та рефакторинку до ПЗ. Програма передбачає вивчення патернів проектування та застосування їх до власних проектів.

У результаті вивчення навчальної дисципліни студент повинен

знати: основні поняття, визначення і проблеми курсу; вимоги до постановки основних методів рефакторінгу та чистого коду; призначення й особливості застосування основних патернів проєктування;

вміти: застосовувати чистий код для розв'язування задач; реалізувати патерни на сучасних задач.

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 С# 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 аналізу засобів інформаційних технологій під час проєктування із розроблення ІТ рішень для розвитку та підтримки та стабільності.
-

3. Програма навчальної дисципліни

Змістовий модуль 1. Refactoring

1. Refactoring: Improving the Design of Existing Code
2. What is Refactoring
3. Code Smells
4. Technical debt
5. Refactoring Techniques
6. Refactoring Techniques: Organization Data
7. How to refactor
8. Analyze Refactoring in projects

Змістовий модуль 2. Design patterns

1. Design patterns. Вступ
2. Benefits of patterns, Classification
3. Design patterns: Creational patterns
4. Design patterns: Structural patterns
5. Design patterns: Behavioral patterns
6. Design patterns: implementation

4. Структура навчальної дисципліни

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 власних 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 год.	6 тиж. семестру
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 проєкту, 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 проєкту	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. Рефакторинг на проєктах. Подальші кроки самовдосконалення.	Лекція	8		

5. Теми семінарських занять

6. Теми практичних занять

7. Теми лабораторних занять

8. Самостійна робота

9. Індивідуальні завдання

№ з/п	Назва теми	Кількість годин
1	<i>Робота над рефакторингом власних проєктів</i>	16
2	<i>Створення проєкту із використанням патернів проєктування</i>	16
	Разом	32

10. Методи навчання

Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи (дискусія).

11. Методи контролю

Поточний контроль здійснюється шляхом проведення усного опитування та написання письмових звітів по виконаних проєктних роботах. У кінці курсу проводиться екзамен.

12. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота		Підсумковий тест (екзамен)	Сума
Змістовий модуль 1	Змістовий модуль 2		
25	25	50	100

Шкала оцінювання: національна та ЄКТС

Оцінка ЄКТС	Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
A	90 – 100	відмінно	зараховано
B	81-89	добре	
C	71-80		
D	61-70	задовільно	
E	51-60		
FX	21-50	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
F	0-20	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

* кількість балів для оцінок «незадовільно» (FX і F) визначається Вченими радами факультетів (педагогічними радами коледжів).

13. Методичне забезпечення

1. Head First Design Patterns: A Brain-Friendly Guide: Building Extensible and Maintainable Object-Oriented Software Paperback – 5 Jan. 2021
2. Refactoring: Improving the Design of Existing Code (Addison-Wesley Signature Series (Fowler)) Hardcover – Illustrated, 2 Jan. 2019

14. Рекомендована література

Основна

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.

Допоміжна

1. Domain-Driven Design: Tackling Complexity in the Heart of Software Hardcover – Illustrated, 11 Sept. 2013
12. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions (Addison-Wesley Signature Series (Fowler)) Hardcover – 23 Oct. 2013.

15. Інформаційні ресурси

1. Internet – джерела.
2. Наукова бібліотека Львівського національного університету імені Івана Франка (<https://www.lnulibrary.lviv.ua/to-users-2/paid-services/internet/>).
3. Львівська національна наукова бібліотека України імені Василя Стефаника (<https://www.lsl.lviv.ua/index.php/uk/elektronni-resursyl/>).