Затверджено Вченою Радою ОНУ
імені І.І. Мечникова
від "20"грудня 20 16 р. № 4

Одеський національний університет імені І.І. Мечникова
Кафедра математичного забезпечення комп'ютерних систем

"ЗАТВЕРДЖУЮ"

Проректор _____

"_____" _____ 20___ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК 22 «Інженерія програмного забезпечення»
(назва навчальної дисципліни)

Рівень вищої освіти _____бакалавр_____

Спеціальність _____126 – Інформаційні системи та технології_____
(код і назва спеціальності (тей)

Факультет _____математики, фізики та інформаційних технологій_____
(назва інституту, факультету)

2021 – 2022

Робоча програма складена на основі навчальної програми з дисципліни «Інженерія програмного забезпечення».

Розробник:
Пенко В.Г., к.т.н., доцент кафедри МЗКС

Робоча програма затверджена на засіданні кафедри математичного забезпечення комп'ютерних систем

Протокол № 1 від. " 30 " 08 2021 р.

Завідувач кафедри _____ (Є.В. Малахов)
(підпис) (прізвище та ініціали)

Схвалено навчально-методичною комісією (НМК) _з ІТ факультету МФІТ_

Протокол № 1 від. " 6 " 09 2021 р.

Голова НМК _____ (О.В. Савастру)
(підпис) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____"_____20__ р.

Завідувач кафедри _____ (_____)
(підпис) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____"_____20__ р.

Завідувач кафедри _____ (_____)
(підпис) (прізвище та ініціали)

1. Опис навчальної дисципліни

<table>
<tr><td rowspan="2">Найменування показників</td><td rowspan="2">Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td>денна форма навчання</td><td>заочна форма навчання</td></tr>
<tr><td rowspan="15">Загальна кількість:

кредитів – 4

годин – 120

залікових модулів – 1

змістових модулів – 2

ІНДЗ* – _____
(вид завдання)</td><td>Галузь знань
12 – Інформаційні технології
(шифр і назва)</td><td colspan="2">Нормативна</td></tr>
<tr><td rowspan="5">Спеціальність
126 – Інформаційні системи та технології
(шифр і назва)</td><td colspan="2">Рік підготовки:</td></tr>
<tr><td>3</td><td></td></tr>
<tr><td colspan="2">Семестр</td></tr>
<tr><td>1</td><td></td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td rowspan="9">Рівень вищої освіти:
бакалавр</td><td>30 год.</td><td></td></tr>
<tr><td colspan="2">Практичні, семінарські</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Лабораторні</td></tr>
<tr><td>30 год.</td><td></td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>60 год.</td><td></td></tr>
<tr><td colspan="2">Індивідуальні завдання:</td></tr>
<tr><td colspan="2">Вид контролю: екзамен</td></tr>
</table>

* – за наявності

2. Мета та завдання навчальної дисципліни

Метою курсу є освоєння теоретичних принципів і практичних навичок використання сучасних технологій для розробки програмних систем середнього та великого масштабу контрольованого якості в умовах, що змінюються вимог і групової розробки.

Завдання:

– - знайомство з передісторією виникнення сучасного поняття програмної інженерії;

– - вивчення основних нормативних документів і методик оцінки зрілості процесів розробки програмного забезпечення;

– - вивчення принципів проектування програмних систем;

– - освоєння основних патернів об'єктно-орієнтованого проектування;

– - освоєння основних принципів і практик методики гнучкої розробки програм.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей**:

1) загальних:

КЗ 12. Здатність застосовувати знання психологічних закономірностей інформаційної взаємодії людини з технічними пристроями і програмним забезпеченням комп'ютерних систем, знання основ мікроергономіки та ергономічних стандартів при створенні інформаційних система для забезпечення комфортної, безпечної та ефективної роботи користувачів

2) спеціальних (фахових):

КС 6. Здатність використовувати сучасні інформаційні системи та технології (виробничі, підтримки прийняття рішень, інтелектуального аналізу даних та інші), методики й техніки кібербезпеки під час виконання функціональних завдань та обов'язків.

КС 12. Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).

Програмні результати навчання:

ПР 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПР 4. Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.

ПР 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.
ПР 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.
ПР 8. Застосовувати правила оформлення проектних матеріалів інформаційних систем та технологій, знати склад та послідовність виконання проектних робіт з урахуванням вимог відповідних нормативно-правових документів для запровадження у професійній діяльності.
ПР 9. Здійснювати системний аналіз архітектури підприємства та його ІТ-інфраструктури, проводити розроблення та вдосконалення її елементної бази і структури.
Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен
знати*: основні причини, що ускладнюють розробку програмного забезпечення, основні моделі життєвого циклу програмного забезпечення, принципи об'єктно-орієнтованого проектування програмного забезпечення.*
вміти*: застосовувати патерни об'єктно-орієнтованого проектування при розробці програм; здійснювати розробку у відповідності з методикою розробки через тестування; здійснювати рефакторинг програмного коду; використовувати системи контролю версій програмного продукту.*

3. Зміст навчальної дисципліни

Змістовний модуль 1 Основні поняття програмної інженерії
Тема 1. Визначення та проблематика програмної інженерії.
Література: [1, 7, 8].
Тема 2. Шаблони проектування.
Література: [2, 10, 13].
Тема 3. Мова UML, основні типи діаграм.
Література: [4, 15].
Тема 4. ЖЦ ПЗ - призначення та структура, процес розробки програмного забезпечення
Література: [1, 14].
Тема 5. Моделі ЖЦ ПЗ: визначення та класифікація. Водопадна модель та спіральна ітераційна модель.
Література: [1, 8].
Тема 6. Модель CMM.
Література: [1].

Змістовний модуль 2 Гнучка розробка програмного забезпечення
Тема 1. Маніфест гнучкої розробки програм : система цінностей та принципи.

Література: [2, 11].
Тема 2. Практики, що використовуються в гнучкому процесі розробки.
Література: [2, 11].
Тема 3. Розробка через тестування (Test Driven Development).
Література: [2, 16].
Тема 4. Використання систем контролю версій
Література: [6, 12].
Тема 5. Рефакторинг.
Література: [2, 5, 9].
Тема 6. Ознаки погано спроектованої програми. Роль принципів і шаблонів об'єктно-орієнтованого проектування в усуненні цих ознак.
Література: [2].
Тема 7. SOLID-принципи проектування.
Література: [2].

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Денна форма					
	Форма контролю	Усього	у тому числі			
			л	п	лаб	ср
1	2	3	4	5	6	7
Змістовний модуль 1. Основні поняття програмної інженерії.						
Тема 1.Визначення та проблематика програмної інженерії.	КО	6	2			4
Тема 2. Шаблони проектування.	КО	12	4		3	5
Тема 3. Мова UML, основні типи діаграм.	КО	11	3		3	5
Тема 4. ЖЦ ПЗ - призначення та структура, процес розробки програмного забезпечення	КО	10	2		3	5
Тема 5. Моделі ЖЦ ПЗ: визначення та класифікація. Водопадна модель та спіральна ітераційна модель.	КО	10	2		3	5
Тема 6. Модель CMM.	КО	10	2		3	5
Змістовний модуль 2. Гнучка розробка програмного забезпечення.						
Тема 1. Маніфест гнучкої розробки програм : система цінностей та принципи.	КО	6	2			4
Тема 2. Практики, що використовуються в гнучкому процесі розробки.	КО	10	3		3	4
Тема 3. Розробка через тестування (Test Driven Development).	КО	9	2		3	4
Тема 4. Використання систем контролю версій	КО	9	2		3	4
Тема 5. Рефакторинг.	КО	10	2		3	5
Тема 6. Ознаки погано спроектованої програми. Роль принципів і шаблонів об'єктно-орієнтованого проектування в усуненні цих ознак.	КО	7	2			5
Тема 7. SOLID-принципи проектування.	КО	10	2		3	5
Всього годин		**120**	**30**		**30**	**60**

Форма контролю: **КО** –контрольне опитування (поточне)

5. Теми семінарських занять

Немає

6. Теми практичних занять

Немає

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Реалізація додатків з використанням патернів об'єктно-орієнтованого проектування.	10
2	Розробка додатків з використанням практик рефакторингу та TDD	10
3	Розробка додатків з використанням системи контролю версій	10
	Разом	**30**

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Додаткові патерни обєктно-орієнтованого проектування.	15
2	Використання розширеного асортименту ре факторингу.	15
3	Використання патерну MVC	15
5	Командна робота з використанням системі контролю версій.	15
	Разом	**76**

До самостійної роботи відноситься:
[1] – підготовка до лекцій та лабораторних занять;

9. Індивідуальне навчально-дослідне завдання

Немає

10. Методи навчання

Лекції з використанням мультимедійного презентаційного матеріалу.

11. Методи контролю

Підсумковим контролем з дисципліни є екзамен.

Критерії оцінювання на підсумковому контролі:

Екзаменаційний квиток по дисципліні складається з двох частин: теоретичної та практичної. Мінімальна кількість балів, зараховуються як позитивний результат, дорівнює 60 (за 100-бальною шкалою).

Бали розподіляються наступним чином: 60 балів - теоретична частина та 40 балів - практична.

Теоретична частина містить 2 питання, практична - 1 питання.

За бездоганну відповідь на кожне теоретичне питання студент отримує - 30 балів. При цьому відповідь вважається бездоганною, якщо студент повністю розкрив

сутність питання, послідовно і логічно його доповів, навів приклади, проілюстрував відповідь необхідним і достатнім кількістю записів, графіків, формул, схем; зробив посилання на відповідні літературні джерела.
За бездоганне виконання завдання практичної частини студент отримує - 40 балів. Завдання практичної частини іспиту вважається виконаним бездоганно, якщо при його вирішенні отримано правильну відповідь, послідовно і логічно викладено рішення, отримано всі сформульовані в завданні результати.

11.1. **Питання до підсумкового контролю**

1. Життєвий цикл програмного забезпечення. Визначення. Структура.
2. Життєвий цикл програмного забезпечення. Процесс розробки.
3. Моделі ЖЦ ПО. Класифікація та структура.
4. Моделі ЖЦ ПО. Водоспадна та спіральна моделі.
5. Сертифікація і оцінка процесів створення ПО. Модель СММ
6. Шаблони проектування. Визначення, призначення. Елементи опису.
7. Маніфест гнучкої розробки програм : система цінностей.
8. Маніфест гнучкої розробки програм : принципи.
9. Рефакторинг: сутність та основні патерни.
10. Основні властивості програмного забезпечення, що ускладнюють його розробку
11. Гнучке проектування: ознаки погано спроектованої програми.
12. Основні джерела проблем при розробці програмного забезпечення
13. Огляд методик екстремального програмування: Єдина команда, Історії користувача, Короткі цикли, Приймальні тести.
14. Огляд методик екстремального програмування: Безперервна інтеграція, Помірний темп, Відкритий робочий простір.
15. Огляд методик екстремального програмування: Парне програмування, Розробка через тестування, Колективне володіння.
16. Огляд методик екстремального програмування: Гра в планування, Простота, Рефакторінг, Метафора.
17. Розробка через тестування (Test Driven Development)
18. Планування процесу розробки в екстремальному програмуванні.
19. Гнучке проектування: роль принципів і шаблонів об'єктно-орієнтованого проектування.
20. Принцип єдиного обов'язку
21. Принцип відкритості/закритості
22. Принцип підстановки Лісков
23. Принцип інверсії залежності

12. **Розподіл балів, які отримують студенти**

Поточне тестування та самостійна робота													Сума
Змістовий модуль №1						Змістовий модуль № 2							
Т1	Т2	Т3	Т4	Т5	Т6	Т1	Т2	Т3	Т4	Т5	Т6	Т7	
8	8	8	8	8	8	8	8	8	7	7	7	7	100

Т1, Т2 ... – теми змістових модулів.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
85-89	B	добре	
75-84	C		
70-74	D	задовільно	
60-69	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13. Методичне забезпечення

Конспект лекцій в електронному форматі.

14. Рекомендована література

14.1. Основна література

1. Бородкіна І.Л., Бородкін Г.О. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів - Центр навчальної літератури - 2018. - 204 с.
2. Пенко В.Г. Патерни об'єктно-орієнтованого програмування: методичний посібник //В.Г. Пенко / Одеса: Одес. нац. ун-т імені І.І. Мечникова, 2021. – 61 с.
3. М. О. Сидоров М.О. Вступ до інженерії програмного забезпечення - Київ: Видавництво Національного авіаційного університету «НАУ-друк», 2010. - 112с.
4. Martin C. Robert, Martin Micah Agile Principles, Patterns, and Practices in C# - Prentice Hall, 2006. - 768 p.
5. Будай А. Дизайн-патерни — просто, як двері . – Режим доступу: https://andriybuday.com/book
6. Постіл С. Д. UML. Уніфікована мова моделювання інформаційних систем - Ірпінь : Університет державної фіскальної служби України. 2019. - 325 с.
7. Martin Fowler Refactoring: Improving the Design of Existing Code Addison Wesleyu Longman, 2000. - 420 p..
8. Scott Shacon, Ben Straub ProGIT - APress Second Edition, 2012. - 574 pp. (Український переклад в електронному вигляді - http://git-scm.com/book/uk/v2).

14.2. **Допоміжна література**

9. Липаев В.В. Программная инженерия. Методологические основы: Учебник. - - М.: ТЭИС, 2004. – 609 с.
10. G. O'Regan, Concise Guide to Software Engineering, Undergraduate Topics in Computer Science, Springer International Publishing AG 2017. - 327 p.
11. Кериевски, Дж. Рефакторинг с использованием шаблонов. : Пер. с англ. — М.: ООО "И.Д. Вильямс", 2006. — 400 с.: ил.
12. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. - 368 с : ил. (Серия «Библиотека программиста»)
13. Kent Beck, Cynthia Andres Extreme Programming Explained: Embrace Change, Second Edition - Addison Wesley Professional 2004 – 224 p.
14. Фаулер М. – UML. Основы 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с., ил.

15. **Електронні інформаційні ресурси**

15. GitHub. – Режим доступу: https://github.com
16. Э.Фримен, Э. Фримен, К. Сьерра, Б. Бейтс Паттерны проектирования. – СПб.: Питер, 2011. – 656 с.: ил.
17. ДСТУ ISO/IEC 12207:2016 Інженерія систем і програмного забезпечення ПРОЦЕСИ ЖИТТЄВОГО ЦИКЛУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ISO/IEC 12207:2008, IDT) – Режим доступу: http://online.budstandart.com/ru/catalog/doc-page.html?id_doc=71824
18. Иванов Д., Новиков Ф. Моделирование на UML Учебно-методическое пособие. – Режим доступу: https://books.ifmo.ru/file/pdf/722.pdf
19. Kent Beck Test-Driven Development By Example – Режим доступу: http://barbra-coco.dyndns.org/yuri/Kent_Beck_TDD.pdf