МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА
Кафедра математичного забезпечення комп'ютерних систем

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи
_____
_____ 20___ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК 27 «Інженерія програмного забезпечення»
(назва навчальної дисципліни)

Рівень вищої освіти *Перший (бакалаврський)*

Галузь знань *12 – Інформаційні технології*

Спеціальність *123 – Комп'ютерна інженерія*
(код і назва спеціальності (тей)

Освітньо-професійна програма *Комп'ютерна інженерія*
(назва ОПП/ОНП)

ОНУ - 2022

Робоча програма навчальної дисципліни «Інженерія програмного забезпечення»: – Одеса: ОНУ, 2022. – 11с..

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

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

Протокол № 1 від. "25" 08 2022р.

Завідувач кафедри _____ (підпис) (Євгеній МАЛАХОВ)
(Ім'я ПРІЗВИЩЕ)

Погоджено із гарантом ОПП «Комп'ютерна інженерія»

_____ (підпис) (Людмила ВОЛОЩУК)
(Ім'я ПРІЗВИЩЕ)

Схвалено навчально-методичною комісією (НМК) з ІТ факультету МФІТ

Протокол № 2 від. "31" 08 2022р.

Голова НМК _____ (підпис) (Алла РАЧИНСЬКА)
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____" _____ 20__ р.

Завідувач кафедри _____ (підпис) (_____)
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____" _____ 20__ р.

Завідувач кафедри _____ (підпис) (_____)
(Ім'я ПРІЗВИЩЕ)

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 – 6

годин – 180

змістових модулів – 2

ІНДЗ* – розрахунково-графічна робота
(вид завдання)</td><td>Галузь знань
12 – Інформаційні технології
(шифр і назва)</td><td colspan="2">Обов’язкова</td></tr>
<tr><td rowspan="5">Спеціальність
123 – Комп’ютерна інженерія
(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 (5)</td><td></td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td rowspan="8">Рівень вищої освіти:
Перший (бакалаврський)</td><td>36 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>54 год.</td><td></td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>90 год.</td><td></td></tr>
<tr><td colspan="2">Індивідуальні завдання:</td></tr>
<tr><td colspan="2">Вид контролю: іспит</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**: **компетентностей** (згідно ОПП «Інформаційні системи та технології» від **2019** р.):

а) загальних

К31. Здатність до абстрактного мислення, аналізу і синтезу.

К38. Здатність працювати в команді.

К312. Здатність застосовувати базові знання з фундаментальної та прикладної математики в професійній діяльності.

б) (фахових):

КС2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

КС3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

КС5. Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

КС11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

КС14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

Програмні результати навчання:

ПР6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПР7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПР8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПР12. Вміти ефективно працювати як індивідуально, так і у складі команди.
ПР13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.
.

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].
Тема 8. Захист розрахунково-графічної роботи.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		л	п	лаб	ср		л	п	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовний модуль 1. Основні поняття програмної інженерії.										
Тема 1.	8	4			4					
Тема 2	28	4		12	12					
Тема 3.	16	4		6	4					
Тема 4.	10	2		2	4					
Тема 5.	10	2		2	4					
Тема 6.	8	2		2	4					
Змістовний модуль 2. Гнучка розробка програмного забезпечення.										
Тема 1.	6	2			4					
Тема 2.	16	4		4	4					
Тема 3.	14	2		4	6					
Тема 4.	16	2		6	6					
Тема 5.	16	2		6	6					
Тема 6.	10	2			6					
Тема 7.	18	4		6	6					
Тема 8.				4	20					
Всього годин	**180**	36		**54**	**90**					

Форма контролю: **КО** –контрольне опитування (поточне)
КР – контрольна робота

5. Теми семінарських занять

Семінарські заняття не передбачені

6. Теми практичних занять

Семінарські заняття не передбачені

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Реалізація додатків з використанням патернів об'єктно-орієнтованого проектування.	22
2	Розробка додатків з використанням практик рефакторингу та TDD	18
3	Розробка додатків з використанням системи контролю версій	14
	Разом	**54**

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Сучасні аспекти програмної інженерії	4
2	Додаткові патерни обєктно-орієнтованого проектування.	12
3	Додаткові елементи мови UML.	4
4	V-модель життєвого циклу ПЗ	4
5	Поглиблене вивчення стандарту життєвого циклу ПЗ	4
6	Особливості моделі CMMI	4
7	Передісторія виникнення гнучкої розробки ПЗ	4
8	Практичне застосування практик гнучкої розробки.	4
9	Практичне застосування TDD.	6
10	Практичне застосування GIT.	6
11	Використання розширеного асортименту рефакторингу.	6
12	Приклади усунення поганих ознак коду	6
13	Приклади використання SOLID -принципів проектування	6
14	Розрахунково-графічна робота	20
	Разом	**90**

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

8.1. Індивідуальне навчально-дослідне завдання

Розрахунково-графічна робота
Метою розрахунково-графічної роботи (РГР), метою якої є придбання та розвиток практичних навичок самостійного розв’язання завдань, що потребують комплексного використання проектно-технологічних прийомів розробки програмного забезпечення.

Завдання на РГР видається на початку 5 семестру. На її виконання відводиться 12,5 годин індивідуальної роботи студента. Після практичного виконання РГР здобувач оформлює звіт та захищає роботу (за 20 бальною системою). Оцінювання РГР відбувається з урахуванням повноти реалізації постановки задачі, компетентності студента в процесі захисту роботи, змістовність програмної реалізації, обґрунтованості прийнятих рішень, ритмічності у роботі (дотримання строків здачі роботи або її складових частин).

Робота повинна підтвердити опанування студентом дисципліни та прищепити навички самостійної роботи при створенні певних проектів, показати компетентність та творчість мислення, обґрунтованість прийнятих проектних рішень.

Основні складові компоненти постановки завдання, що повинні бути відображені у звіті:

- доречне використання в програмній реалізації 2-х або 3-х різних патернів об'єктно-орієнтованого проектування – здобувач на протязі 4-х тижнів повинен запропонувати та затвердити у викладача власну постановку завдання (предметної області завдання);
- використання системи контролю версій (наприклад, Git) для фіксації етапів розробки у вигляді розгалуженої системи фіксації (декілька гілок);
- використання не менш 5 прийомів рефакторінгу;
- використання практичної методики Test Driven Development у вигляді накопичуваної послідовності тестів;
- діаграмма класів, що відображає проектне рішення.

В процесі виконання РГР здійснюється проміжний контроль (2 проміжних етапи), зміст і терміни яких вказано відповідним графіком. Звіт оформлюється у вигляді пояснювальної записки та складає 20-25 аркушів формату А4 (текстові пояснення, важливі фрагменти програмної реалізації, додатки у вигляді тексту реалізації та результатів виконання програми, діаграма класів). Звіт подається на перевірку не пізніше передостаннього тижня. Захист РГР проводиться на останньому тижні.

9. Методи навчання

Лекції з використанням мультимедійного презентаційного матеріалу.

10. Методи контролю

Підсумковим контролем з дисципліни є іспит.

10.1. 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. **Питання до підсумкового контролю**

1. Життєвий цикл програмного забезпечення. Визначення. Структура.
2. Життєвий цикл програмного забезпечення. Процес 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. Патерни проектування. Визначення, призначення. Елементи опису.
7. Маніфест гнучкої розробки програм : система цінностей.
8. Маніфест гнучкої розробки програм : принципи.
9. Рефакторинг: сутність та основні прийоми.
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	Т8	
6	6	6	8	8	8	6	6	6	6	6	6	6	16	100

Т1, Т2 ... – теми змістових модулів.

Шкала оцінювання: національна та ECTS

Загальна сума балів	Оцінка ECTS	Національна шкала	
90 — 100	A – «відмінно»	5 «відмінно»	«зарахо»
85 — 89	B – «дуже добре»	4 «добре»	
75 — 84	C – «добре»		
70 — 74	D – «задовільно»	3 «задовільно»	
60 — 69	E – «допустимо»		
35 — 59	F – «незадовільно з можливістю повторного складання»	2 «незадовільно»	«незарахо»
0 — 34	FX – «незадовільно з обов’язковим повторним курсом»		

13. **Навчально-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. 2011 – 200с. – 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