

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ЗВ'ЯЗКУ ім. О.С. ПОПОВА**  
(повне найменування вищого навчального закладу)

---

**Кафедра інформаційних технологій**

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**Основи програмної інженерії**

Спеціальність **121 Інженерія програмного забезпечення**

факультет, відділення **Інфокомунікацій та програмної інженерії**

Одеса  
2020 рік

Робоча програма \_\_\_\_\_ Основи програмної інженерії \_\_\_\_\_  
для студентів  
за спеціальністю 121 Інженерія програмного забезпечення

Розробник програми: Трофименко О.Г. \_\_\_\_\_  
\_\_\_\_\_

Робоча програма затверджена на засіданні кафедри *Інформаційних технологій*

Протокол від “ \_\_\_\_\_ ” \_\_\_\_\_ 2020 року № \_\_\_\_\_

Завідувач кафедри Інформаційних технологій

\_\_\_\_\_ ( \_\_\_\_\_ )

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 року

### 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 3	Галузь знань <i>12 Інформаційні технології</i> (шифр і назва)	Нормативна	
	Напрямок підготовки <i>121 Інженерія програмного забезпечення</i> (шифр і назва)		
Семестрів – 1	Спеціальність (професійне спрямування): _____	<b>Рік підготовки:</b>	
Змістових модулів – 2		1	
Індивідуальне науково-дослідне завдання _____ (назва)		<b>Семестр</b>	
Загальна кількість годин – 90		1.2	
Тижневих годин для денної форми навчання: аудиторних – 2 самостійної роботи студента – 2,5	Освітньо-кваліфікаційний рівень: <i>Бакалавр</i>	<b>Лекції</b>	
		14 год.	
		<b>Практичні, семінарські</b>	
		14 год.	
		<b>Лабораторні</b>	
		12 год.	
		<b>Самостійна робота</b>	
50 год.			
<b>Індивідуальні завдання:</b>			
Вид контролю: Залік.			

## 2. Анотація дисципліни

Навчальна дисципліна «Основи програмної інженерії» призначена для здобувачів вищої освіти, що навчаються за освітньо-кваліфікаційним рівнем «бакалавр» галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». Ця навчальна дисципліна в числі інших складає теоретично-практичну основу сукупності компетентностей, що формують профіль фахівця в галузі проєктування програмного забезпечення.

Ця навчальна дисципліна знайомить майбутніх програмістів з основними аспектами виробництва програмного забезпечення від початкових стадій створення специфікації до підтримки системи після здачі в експлуатацію.

Програма навчальної дисципліни складається з таких змістових модулів:

1. Інженерія програмного забезпечення.
2. Етапи розроблення програмного забезпечення.

## 3. Мета та завдання навчальної дисципліни

**Предметом** навчальної дисципліни є основні поняття програмної інженерії та методи створення надійного, якісного програмного забезпечення, що відповідає вимогам замовника.

**Метою** викладання навчальної дисципліни “Основи програмної інженерії” є формування у майбутніх програмістів розуміння важливості застосування принципів програмної інженерії (software engineering) при розробці програмного забезпечення (ПЗ), вивчення базових понять програмної інженерії та набуття навичок її використання для реалізації нескладних програмних систем. У курсі розглядається увесь спектр процесів, що ведуть до створення ПЗ: від початкової розробки системних вимог і далі через проєктування, безпосереднє програмування і атестацію до модернізації програмних систем. Крім того, курс знайомить студентів із соціальними та професіональними питаннями програмування, зокрема з професійною та етичною відповідальністю фахівця з програмної інженерії.

**Основними завданнями** дисципліни є засвоєння студентами основ програмної інженерії, ключових понять, методів і засобів програмної інженерії як діяльності, спрямованої на створення програмних систем, що відповідають потребам замовників, застосування інженерами теоретичних методів і засобів розробки ПЗ, стандартів, правил і методик управління розробкою ПЗ, а також питань проведення інженерних оцінок якості ПЗ.

В результаті успішного засвоєння навчальної дисципліни студент матиме змогу продемонструвати такі результати навчання:

**знати:** основні поняття інженерії програмного забезпечення (ПЗ); етапи розроблення ПЗ; основні вимоги державних та міжнародних стандартів, що регламентують створення ПЗ; життєвий цикл ПЗ; моделі життєвого циклу програмних забезпечення; методи об'єктного аналізу і моделювання програмних систем; основні показники якісного ПЗ; моделі якості та надійності програмних систем;

**вміти:** здійснювати аналіз вимог до ПЗ, формулювати вимоги до нескладних програмних продуктів; розробляти UML-діаграми різних типів при моделюванні різних аспектів ПЗ на різних етапах життєвого циклу інформаційної систе-

ми; аналізувати інформаційні системи та класифікувати їх за різними критеріями.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **компетентностей**:

**загальних:**

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

K05. Здатність вчитися і оволодівати сучасними знаннями.

K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

**фахових:**

K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

K26. Здатність до алгоритмічного та логічного мислення.

Результати навчання даної дисципліни деталізують такі **програмні результати навчання**:

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.

ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

На вивчення навчальної дисципліни відводиться 120 годин / 3 кредити ECTS.

#### 4. Інформаційний обсяг навчальної дисципліни

**Змістовий модуль 1.** Інженерія програмного забезпечення.

*Тема 1.* Програмна інженерія як дисципліна. SWEBOOK – Керівництво до зводу знань з програмної інженерії.

*Тема 2.* Програмне забезпечення. Технологія програмування в історичному аспекті. Основні поняття і визначення. Класифікація програмного забезпечення.

*Тема 3.* Процес створення програмного забезпечення.

**Змістовий модуль 2.** Етапи розроблення програмного забезпечення.

*Тема 4.* Загальна структура і зміст життєвого циклу (ЖЦ) програмних систем у стандарті ISO/IEC 12207:2002. Формування прикладних моделей ЖЦ. Типи моделей ЖЦ.

*Тема 5.* Проектування програмного забезпечення.

*Тема 6.* Визначення вимог до програмних систем.

*Тема 7.* Прототипування.

*Тема 8.* Методи об'єктного аналізу і моделювання.

*Тема 9.* Моделі якості та надійності програмних систем.

#### 5. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	денна форма				
	усього	у тому числі			
л		п	лаб	с.р	
<i>1</i>	2	3	4	5	7
<b>Модуль 1 (кредитів – 3)</b>					
<b>Змістовий модуль 1. Інженерія програмного забезпечення</b>					
<i>Тема 1.</i> Програмна інженерія як дисципліна. SWEBOOK – Керівництво до зводу знань з програмної інженерії.	8	2	2		4
<i>Тема 2.</i> Програмне забезпечення. Технологія програмування в історичному аспекті. Основні поняття і визначення. Класифікація програмного забезпечення.	5	2			3
<i>Тема 3.</i> Процес створення програмного забезпечення.	5		2		3
<i>Разом за змістовним модулем 1</i>	18	4	4		10
<b>Змістовий модуль 2. Проектування програмного забезпечення</b>					
<i>Тема 4.</i> Загальна структура і зміст життєвого циклу (ЖЦ) програмних систем у стандарті ISO/IEC 12207:2002. Формування прикладних моделей ЖЦ. Типи моделей ЖЦ.	9	2	2		5
<i>Тема 5.</i> Проектування програмного забезпечення.	18	2	2	4	10
<i>Тема 6.</i> Визначення вимог до програмних систем.	14	2		4	8
<i>Тема 7.</i> Прототипування.	5		2		3
<i>Тема 8.</i> Методи об'єктного аналізу і моделювання.	13	2	2	2	7
<i>Тема 9.</i> Моделі якості та надійності програмних систем.	13	2	2	2	7
<i>Разом за змістовним модулем 2</i>	72	10	10	12	40
<b>Усього годин</b>	<b>90</b>	<b>14</b>	<b>14</b>	<b>12</b>	<b>50</b>

## 6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Процес створення програмного забезпечення	2
2	Планування проєкту із застосуванням Projectlibre	2
3	Прототипування	2
4	UML- діаграми, основи побудови	2
5	Системи контролю версій.	2
6	Система контролю версій GIT. Робота в консолі	2
7	Організація тестування проєкту	2

## 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Розробка інтелектуальної карти за допомогою програми FreeMind	2
2	Створення проєкту та календарне планування робіт у ProjectLibre	2
3	Створення діаграми варіантів використання (use case diagram)	2
4	Створення діаграми послідовностей (sequence diagram)	2
5	Створення діаграми класів (class diagram)	2
6	Робота з системою контролю версій GIT	2

## 8. Самостійна робота

№ з/п	Види роботи	Кількість годин
1	Проробка лекцій	7
2	Вивчення додаткового матеріалу до лекцій	7
3	Підготовка до практичних занять	14
4	Підготовка до лабораторних робіт	12
Разом:		50

## 9. Методи навчання

Мультимедійні лекції, демонстрування, викладання-пояснення, практичні заняття та лабораторні роботи, запитання-бесіда, метод проблемно-орієнтовного навчання, самостійне спостереження, ілюстрування.

## 10. Методи контролю

Контроль рівня виконавських умінь для перевірки практичної підготовки, усне опитування, тестування.

## 11. Розподіл балів, які отримують студенти

Зміст. модуль № 1				Зміст. модуль № 2							Загальний тест	Сума
t1	t2	t3	тест	t4	t5	t6	t7	t8	t9	тест		
4	3	3	10	4	14	14	2	8	8	10	20	100

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	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

### 12.Методичне, технічне й програмне забезпечення /обладнання

Презентації лекцій, комплекс навчально-методичного забезпечення дисципліни. Для навчального процесу потрібно: браузер, редактор перегляду pdf-файлів Foxit Reader (<https://www.foxitsoftware.com/pdf-reader/>). В якості редактора для розробки UML-діаграм можна використовувати безкоштовний продукт StarUML. Також можна використовувати онлайн середовища, наприклад, Diagrams.net (<https://app.diagrams.net>), Gliffy (<https://www.gliffy.com>), Umbrello (<https://umbrello.kde.org/>), Cacao (безкоштовна версія) (<https://cacao.com>), GenMyModel (безкоштовна версія) (<https://www.genmymodel.com/uml>), Creately (безкоштовна версія) (<https://creately.com>), Lucidchart (безкоштовна версія) (<https://www.lucidchart.com>).

### 13.Рекомендована література

1. IEEE Guide to the Software Engineering Body of Knowledge - SWEBOK®, 2004.
2. Лавріщева К.М. Програмна інженерія. К. 2008. 319 с.
3. Janarthanam S. Hands-On Chatbots and Conversational UI Development. Birmingham-Mumbai: Packt Publishing, 2017. 373 p.
4. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. СПб: Питер, 2019. 351 с.
5. Мартин Р. Чистый код : создание, анализ и рефакторинг. СПб.: Питер, 2018. 464 с.
6. Сонмез, Д. Путь программиста. Человек эпохи IT; пер. с англ. Е. Зазноба. СПб.: Питер, 2016. 448 с.
7. Соммервиль И. Инженерия программного обеспечения. М.: 2002. 624 с.
8. Карпенко В. Введение в программную инженерию. М.: 2005.
9. Орлов С. Технологии разработки программного обеспечения. СПб.: Питер, 2002. 464 с.
10. Буч Г., Рамбо Дж., Джекобсон А. UML. Руководство пользователя. М.: 2005. 257 с.
11. Бехтерев С. Мейнд-менеджмент. М.: Альпина Паблишер, 2009. 308 с.
12. Основи програмної інженерії: навч. посібник / Є. О. Зайцев. К.: КНТЕУ, 2017. 423 с.



## Інформаційні ресурси

1. <http://www.it.onat.edu.ua/> – Сайт кафедри інформаційних технологій, на якому розміщено робочі матеріали з курсу.
2. <https://www.computer.org/web/swebok> – IEEE Computer Society – SWEBOOK.
3. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series. <http://sites.computer.org/ccse/SE2004Volume.pdf>
4. IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. – [http://standards.ieee.org/reading/ieee/std\\_public/description/se/610.12-1990\\_desc.html](http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html).
5. Липав В. Стандарты, регламентирующие жизненный цикл сложных программных комплексов. – <http://www.pcweek.ru/year1998/N24/CP1251/Reviews/chapt1.htm>
6. Boehm В. A Spiral Model of Software Development and Enhancement. – IEEE Computer. – <http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral%20Model%20Boehm.pdf>
7. Арчибальд Р. Модели жизненного цикла высокотехнологичных проектов. – <http://www.pmpofy.ru/content/rus/107/1073-article.asp>.
8. Кони Бюргер. От ремесла к науке: поиск основных принципов разработки ПО. – <http://www.interface.ru/fset.asp?Url=/rational/science.htm>.
9. Glossary of Software Engineering terms. – <http://www.shellmethod.com/refs/seglossary.pdf>.
10. IEEE-CS/ACM Software Engineering Ethics and Professional Practices. – [http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs\\_level1&path=ieeecs/content&file=ethics.xml&xsl=generic.xsl&](http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs_level1&path=ieeecs/content&file=ethics.xml&xsl=generic.xsl&).
11. Електронний конспект лекцій.
12. Методичні вказівки до виконання лабораторних робіт та практичних занять в електронному вигляді.