

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ПРОМИСЛОВОЇ АВТОМАТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО
УНІВЕРСИТЕТУ»**

Циклова комісія комп'ютерних наук та інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

В. о. директора ФКПАІТ ОНТУ

_____ Ольга ЄПУР

___. __. 2025 року

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

обов'язкова

Освітньо-професійна програма «Комп'ютерні науки» _____

Код та найменування спеціальності 122 Комп'ютерні науки _____

Шифр та найменування галузі знань 12 Інформаційні технології _____

Мова навчання українська _____

Розроблено та забезпечується: цикловою комісією Комп'ютерних наук та інженерії програмного забезпечення ВСП «Фаховий коледж промислової автоматики та інформаційних технологій Одеського національного технологічного університету»

Розробник:

Юлія БУРМАКІНА, викладач вищої кваліфікаційної категорії ФКПАІТ ОНТУ

Розглянуто та схвалено на засіданні циклової комісії Комп'ютерних наук та інженерії програмного забезпечення

Протокол № 1 від 27. 08. 2025 р.

Голова циклової комісії

(підпис)

Тетяна КОСТИРЕНКО

(Власне ім'я, ПРІЗВИЩЕ)

Гарант освітньо-професійної програми

(підпис)

Наталія ШВЕЦЬ

(Власне ім'я, ПРІЗВИЩЕ)

Розглянуто та схвалено Методичною радою ФКПАІТ ОНТУ

Протокол №__ від __. __.20__р.

Голова Методичної ради ФКПАІТ ОНТУ

(підпис)

Ірина ГЕНЕРАЛОВА

(Власне ім'я, ПРІЗВИЩЕ)

1. Пояснювальна записка

Вступ

Програма вивчення навчальної дисципліни «Конструювання програмного забезпечення» складена відповідно до освітньо-професійної програми підготовки фахового молодшого бакалавра спеціальності 122 «Комп'ютерні науки».

Предмет вивчення

Предметом вивчення навчальної дисципліни є основи конструювання програмного забезпечення та уніфікована мова моделювання програмного забезпечення UML.

Міждисциплінарні зв'язки

Попередні - «Основи програмної інженерії», «Об'єктно-орієнтоване програмування», «Бази даних»; послідовні – «Тестування програмного забезпечення», «Технології захисту інформації».

Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни «Конструювання програмного забезпечення» є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методологій та технологій розробки, тестування та експлуатації програмних продуктів, та для моделювання і проектування інформаційних систем, документування та маркетингу програмного забезпечення. Для моделювання та проектування розглядається уніфікована мова моделювання UML.

Основними завданнями вивчення дисципліни «Конструювання програмного забезпечення» є:

1. **Засвоєння основ UML**, зокрема опанування базових понять та елементів уніфікованої мови моделювання, таких як діаграми варіантів використання, діаграми класів, діаграми послідовностей та інші.
2. **Моделювання вимог засобами UML**, що передбачає застосування діаграм варіантів використання для формалізованого та наочного опису функціональних вимог до програмного продукту.
3. **Проектування програмних систем із використанням діаграм класів і співпраці**, що дозволяє визначати структуру системи та характер взаємодії між її компонентами.
4. **Моделювання сценаріїв взаємодії**, шляхом використання діаграм послідовностей для відображення обміну повідомленнями між об'єктами системи в різних сценаріях її функціонування.
5. **Опис поведінки системи та її компонентів**, через застосування діаграм станів для представлення переходів між різними станами об'єктів.
6. **Використання UML для документування програмних проєктів**, з метою створення структурованої, зрозумілої та повної технічної документації.
7. **Застосування UML у практичній проєктній діяльності**, що передбачає реалізацію набутих знань під час розроблення навчальних програмних проєктів.

Метою реалізації зазначених завдань є формування у здобувачів освіти навичок застосування UML як дієвого засобу аналізу, проектування та документування програмного забезпечення, що забезпечує підвищення

ефективності командної взаємодії та створення якісних програмних продуктів у практичних умовах сучасної ІТ-індустрії.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з літературою з питань базисної технології створення програмних продуктів як сукупності процесів розроблення програмних систем.

Компетентності та результати навчання

У результаті вивчення навчальної дисципліни «Конструювання програмного забезпечення» здобувач освіти отримує наступні програмні компетентності та програмні результати навчання, які визначені в Стандарті фахової передвищої освіти із спеціальності 122 «Комп'ютерні науки» (<https://mon.gov.ua/static-objects/mon/sites/1/Fakhova%20peredvyshcha%20osvita/Zatverdzeni.standarty/2021/11/30/122-Kompyuterni.nauky.30.11.pdf>) та освітньо-професійній програмі «Комп'ютерні науки» (<https://promavt.od.ua/storage/uploads/DXbHAM7yoQmTpSH4pc5TUAPTakBhyyPfTSBhLInL.pdf>) підготовки фахових молодших бакалаврів.

Загальні компетентності:

ЗК03. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК04. Здатність застосовувати знання в практичних ситуаціях.

ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК08. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні (фахові, предметні) компетентності:

СК02. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.

СК03. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища.

СК04. Здатність здійснювати проектування та розробку програмного забезпечення.

СК08. Здатність застосовувати сучасні методи, технології та інструментальні засоби проектування та створення програмних систем та їх супроводження.

СК11. Здатність застосовувати методи та техніки тестування програмного забезпечення впродовж життєвого циклу розробки програмних систем.

СК14. Здатність проектувати ІС у відповідності до профілю підготовки по видах забезпечення.

Програмні результати навчання:

РН10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення.

РН12. Знати основні принципи функціонування системного та прикладного програмного забезпечення.

РН15. Розробляти супровідну документацію на різних етапах процесу життєвого циклу розробки програмного забезпечення.

2 Інформаційний обсяг навчальної дисципліни

2.1 Тематичний план

№	Назва змістових модулів і тем
Змістовий модуль 1: Конструювання програмного забезпечення.	
Тема 1	Вступ до конструювання програмного забезпечення. Задачі конструювання програмного забезпечення.
Тема 2	Планування конструювання програмного забезпечення.
Тема 3	Проблеми розробки складних програмних систем
Тема 4	Проектування при конструюванні.
Тема 5	Життєвий цикл програмного забезпечення. Моделі життєвого циклу програмного забезпечення.
Тема 6	Основні етапи розробки програмного забезпечення.
Змістовий модуль 2: Уніфікована мова моделювання UML.	
Тема 7	Мова моделювання UML. Структура і компоненти мови UML.
Тема 8	Основи об'єктно-орієнтованого представлення програмних систем. Об'єкти. Загальна характеристика.
Тема 9	Діаграма прецедентів (діаграма варіантів використання або Use case Diagram).
Тема 10	Діаграма класів.
Тема 11	Діаграма схем станів.
Тема 12	Діаграма діяльності.
Тема 13	Діаграма послідовностей дій.
Тема 14	Діаграма співпраці (кооперацій).
Тема 15	Діаграма компонентів.
Тема 16	Діаграма розгортання.
Разом: 210 годин	

2.2 Зміст дисципліни

Змістовий модуль 1: Конструювання програмного забезпечення.

Тема 1. Вступ до конструювання програмного забезпечення. Задачі конструювання програмного забезпечення.

Поняття конструювання програмного забезпечення. Головні компоненти конструювання програмного забезпечення. Область знань конструювання програмного забезпечення. Стилi конструювання програмного забезпечення. Моделі конструювання, планування та внесення змін.

Тема 2. Планування конструювання програмного забезпечення

Постановка завдання. Визначення основних термінів (потреби, вимога, формалізація постановки завдання, аналіз вимог, специфікація, управління вимогами). Планування конструювання. Декомпозиція складних об'єктів. Покрокова деталізація. Блочно-ієрархічний підхід конструювання програмних систем. Локальна оптимізація. Рівні та аспекти проектування. Процеси технологічного циклу конструювання ПЗ. Методи керування і планування проектом. Діаграмна схема обліку часу Ганта. Мережні методи планування і керування. Види планів організації проекту.

Тема 3. Проблеми розробки складних програмних систем

Фактори, що збільшують складність розробки програмних систем. Блочно-ієрархічний підхід конструювання програмних систем. Декомпозиція складних об'єктів. Покрокова деталізація. Локальна оптимізація.

Рівні та аспекти проектування. Процеси технологічного циклу конструювання ПЗ. Методи керування і планування проектом.

Тема 4. Проектування при конструюванні.

Проблеми, пов'язані з проектуванням програмного забезпечення. Бажані характеристики проекту. Рівні проектування. Компоненти проектування.

Методики проектування.

Тема 5. Життєвий цикл програмного забезпечення. Моделі життєвого циклу програмного забезпечення.

Поняття життєвого циклу програмного забезпечення. Стадії створення програмного забезпечення. Поняття моделі життєвого циклу програмного забезпечення. Каскадна, ітеративна та спіральна моделі життєвого циклу програмного забезпечення.

Тема 6. Основні етапи розробки програмного забезпечення

Етапи розробки програмного забезпечення. Питання, які вирішуються на кожному з етапів розробки програмного забезпечення.

Змістовий модуль 2: Уніфікована мова моделювання UML.

Тема 7. Мова моделювання UML. Структура і компоненти мови UML.

Загальні принципи моделювання систем. Словник UML. Поняття *сутності* в UML. Види сутностей в UML. Поняття *класу*, *інтерфейсу*, *кооперації*, *прецеденту*, *компоненту* та *вузла*. Відношення в UML.

Поняття *діаграми* в UML.

Тема 8. Основи об'єктно-орієнтованого представлення програмних систем. Об'єкти. Загальна характеристика. Поняття *декомпозиції*. Принципи об'єктно-орієнтованої декомпозиції: абстрагування, інкапсуляція, модульність й ієрархічна організація. Загальна характеристика об'єктів. Поняття об'єкт, індивідуальність, стан, поведінка. Види операцій. Види відношень між об'єктами: зв'язок та агрегація.

Тема 9. Діаграма прецедентів (діаграма варіантів використання або діаграма Use case).

Діаграма прецедентів. Актори та варіанти використання на діаграмі прецедентів. Інтерфейси та примітки на діаграмі прецедентів. Зв'язки між елементами діаграми прецедентів: відношення асоціації, відношення узагальнення, відношення включення, відношення розширення.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 10. Діаграма класів

Діаграма класів. Переваги та недоліки класів. Поняття класу. Компоненти діаграми класів. Елементи діаграми класів: класи, інтерфейси, залежності, узагальнення та асоціації. Поняття імені, атрибуту та операції(методу) класу.

Квантор видимості та значення, які він може приймати.

Відношення між класами: асоціація, залежність, узагальнення, реалізація, агрегація, композиція.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 11. Діаграма схем станів.

Моделювання поведінки програмної системи. Поняття *автомату* та *взаємодії*. Призначення діаграми схем станів.

Обов'язкові умови формалізму звичайного автомату.

Поняття стану. Зображення стану на діаграмі. Секції, на які може бути розподілений стан: ім'я стану та список внутрішніх дій або переходів.

Поняття *мітки дій* та значення, які вони можуть приймати.

Початковий та кінцевий стан на діаграмі станів.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 12. Діаграма діяльності.

Різниця діаграми діяльності від діаграми станів. Елементи, які використовуються на діаграмі діяльності. Поняття *діяльності* та *стану* на діаграмі діяльності. Переходи на діаграмі.

Призначення доріжок на діаграмі діяльності. Об'єкти на діаграмі.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 13. Діаграма послідовностей дій.

Призначення діаграми послідовностей дій. Об'єкти на діаграмі. Призначення понять *лінія життя* та *фокус керування*. Види повідомлень.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 14. Діаграма співпраці (кооперацій).

Цілі діаграми співпраці. Базові поняття. Кооперація. Об'єкти на діаграмі. Графічне зображення об'єктів на діаграмі кооперації. Стереотипи зв'язків. Повідомлення. Формат запису повідомлень.

Практичні приклади побудови діаграм кооперації

Тема 15. Діаграма компонентів.

Діаграма компонентів. Елементи діаграми компонентів: компоненти та інтерфейси. Різновиди компонентів. Особливості компонентів. Стандартні стереотипи, передбачені в UML для компонентів.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

Тема 16. Діаграма розгортання.

Діаграма розгортання(розміщення). Елементи діаграми розгортання: вузли, відношення залежності, відношення асоціації.

Графічна нотація діаграми.

Навики побудови діаграми за вибраною предметною областю.

2.3 Перелік практичних робіт з дисципліни

Таблиця 2.2 – Тематичний план практичних робіт

№	Тема практичної роботи
1.	Опис та аналіз предметної області.
2.	Побудова діаграми прецедентів (Use case Diagram)
3.	Побудова діаграми класів (Class Diagram)
4.	Побудова діаграми схем станів (Statechart Diagram)
5.	Побудова діаграми діяльності (Activity Diagram)
6.	Побудова діаграми послідовностей дій (Sequence Diagram)
7.	Побудова діаграми співпраці (кооперації) (Collaboration Diagram)
8.	Побудова діаграми компонентів (Component Diagram)
9.	Побудова діаграми розгортання (Deployment Diagram)
	Всього: 30 годин

2.4 Кількість годин разом за програмою дисципліни

Розподіл дисципліни у годинах									
Курс	I		II		III		IV		Всього
Семестр	1	2	3	4	5	6	7	8	
Повний обсяг часу							210		210
Аудиторні заняття, годин							66		66
із них (кількість годин):									
лекції							36		36
практичні							30		30
Самостійна робота							144		144

3. Інструменти, обладнання та програмне забезпечення, використання яких передбачає дисципліна

1. Безкоштовний онлайн-інструмент для створення діаграм, блок-схем та візуальних схем - <https://app.diagrams.net/>
2. Інструмент для візуального моделювання програмного забезпечення - <https://staruml.io/>
3. Інструмент для побудови UML-схем - <https://creately.com>

4. Інформаційні ресурси

4 Рекомендована література

4.1 Основна література

- Григорович О.Є., Ковальчук В.В. «Моделювання програмних систем засобами UML» - Львів : Видавництво Львівської політехніки, 2012.

- Кунгурцев О.Б. «Конструювання програмного забезпечення. Об'єктно-орієнтований підхід» - навчальний посібник
- Павленко П.М., Федорчук В.А. «UML у проектуванні програмного забезпечення» - К.: НАУ, 2010.
- Жежерун С.М. «Об'єктно-орієнтований аналіз та проектування з використанням UML» - Тернопіль : ТНТУ, 2014.

4.2 Додаткова література

- С. О. Цибульник, К. С. Барандич «Технології розроблення програмного забезпечення частина 1. Життєвий цикл програмного забезпечення», Київ КПІ ім. Ігоря Сікорського 2022
- <http://www.znannya.org> – електронний ресурс
- <http://sites.znu.edu.ua> – електронний ресурс

5. Форма підсумкового контролю

екзамен (VII семестр)

6. Засоби діагностики результатів навчання

Перевірка та оцінювання знань здобувачів освіти може проводитись кількома методами:

1. Оцінювання знань здобувача освіти під час практичних занять.
2. Виконання індивідуальних навчально-дослідних завдань.
3. Захист практичних робіт.
4. Тестування.
5. Проведення поточно-модульного контролю.
6. Проведення заліку.

6.1 Питання для самоконтролю

1. Що описує термін «конструювання програмного забезпечення»?
2. У чому полягає зв'язок конструювання з проектуванням та тестуванням?
3. З чого складається область знань «Конструювання програмного забезпечення»?
4. Які існують стилі конструювання ПЗ? Чим характеризується кожен стиль?
5. Наведіть конкретні завдання, пов'язані з конструюванням. Чому конструювання ПЗ так важливо?
6. Що розуміється під поняттям «Життєвий цикл програмного забезпечення»?
7. Що таке модель життєвого циклу програмного забезпечення? Які бувають моделі?
8. Що таке UML? З чого складається словник UML?
9. Які предмети UML бувають? Що таке структурні предмети? Які існують предмети поведінки?
10. Що таке діаграма?
11. Для чого призначена діаграма варіантів використання? З яких елементів вона складається?

12. Що таке «актор» в термінології UML? Як позначається «актор» на діаграмі варіантів використання? Що на діаграмі варіантів використання зображується еліпсом?
13. Перелічіть відношення, що можуть використовуватися на діаграмі Use Case?
14. Чим відрізняються друг від друга відношення включення й розширення з погляду керування?
15. Які відношення можуть використовуватися для поєднання тільки акторів, акторів та варіантів використання, тільки варіантів використання?
16. Що таке клас та атрибут класу, квантор видимості та операція класу? Як записуються операції?
17. Що таке асоціація?
18. Для чого використовується кратність? Що таке N-арна асоціація?
19. Що таке спадкування та множинне спадкування?
20. Для чого використовують рядки-обмеження? Які ключові слова можна використовувати в якості рядків-обмежень? Що вони означають?
21. Що таке агрегація? Як графічно можна представити агрегацію? В чому різниця між агрегацією та композицією?
22. Що таке залежність? Що таке стереотип залежності? Які стереотипи залежностей вам відомі?
23. Чим відрізняється процедурний потік від асинхронного потоку повідомлень?
24. Як вказується повторення повідомлень? Як показати розгалуження повідомлень?
25. Чим відрізняється процедурний потік від асинхронного потоку повідомлень?
26. Як вказується повторення повідомлень? Як відображається порядок передачі повідомлень у діаграмі послідовності?
27. Коли зручніше застосовувати діаграми послідовності?
28. Як представляється ім'я об'єкта в діаграмі кооперацій? Що таке мультиоб'єкт на діаграмі кооперацій?
29. Що загального в діаграмі послідовності й діаграмі співробітництва? Чим вони відрізняються друг від друга?
30. Що використовують для моделювання поведінки системи?
31. Дайте визначення поняттю автомат? За допомогою чого можна відобразити автомат?
32. Що відображає діаграма схем станів? На виконанні яких обов'язкових умов заснований формалізм діаграм станів?
33. Що таке стан?
34. Дайте визначення поняттю список внутрішніх подій. Що таке подія?
35. Для чого використовується сторожова умова на діаграмі схем станів?
36. В чому різниця діаграми діяльності від діаграми станів?
37. Які елементи використовуються на діаграмі діяльності?
38. Що таке діяльність та стан дії?
39. Для чого необхідні доріжки на діаграмі діяльності?
40. Що відноситься до елементів компонентної діаграми?
41. Що таке компонент? Як задається ім'я компонента?
42. Що таке інтерфейс?
43. Які стандартні стереотипи передбачені в UML для компонентів?
44. Які вершини й ребра утворюють діаграму розміщення?
45. Чим відрізняється вузол від компонента на діаграмі розміщення?