

**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
директор ФКПАІТ ОНТУ
підписано Ольга ЄПУР
30.08.2023 року

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

обов'язкова

Освітньо-професійна програма	<u>Комп'ютерні науки</u>
Код та найменування спеціальності	<u>122 «Комп'ютерні науки»</u>
Шифр та найменування галузі знань	<u>12 «Інформаційні технології»</u>
Мова навчання	<u>українська</u>

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: Тетяна КОСТИРЕНКО, викладач вищої кваліфікаційної категорії ФКПАІТ ОНТУ

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

Протокол №1 від 28.08.2023 р.

Голова циклової комісії

підписано
(підпис)

Тетяна КОСТИРЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Гарант освітньо-професійної програми

підписано
(підпис)

Наталія ШВЕЦЬ
(Власне ім'я, ПРІЗВИЩЕ)

Розглянуто та схвалено Методичною радою ФКПАІТ ОНТУ

Протокол №1 від 29.08.2023 р.

Голова Методичної ради ФКПАІТ ОНТУ підписано Вікторія ОКСАНІЧЕНКО
(підпис)

1. Пояснювальна записка

Вступ

Програма вивчення навчальної дисципліни «Основи програмної інженерії» складена відповідно до Стандарту фахової передвищої освіти із спеціальності 122 Комп'ютерні науки та освітньо-професійної програми «Комп'ютерні науки» підготовки фахових молодших бакалаврів.

Предмет вивчення

Предмет вивчення дисципліни "Основи програмної інженерії" - це комплекс понять, принципів, інструментів та практичних навичок, пов'язаних з процесом аналізу, проектування, розробки та супроводу програмного забезпечення.

Міждисциплінарні зв'язки

Попередні – Основи програмування та алгоритмічні мови, теорія алгоритмів, інженерна і комп'ютерна графіка, об'єктно-орієнтоване програмування, послідовні – конструювання програмного забезпечення, розробка клієнт-серверних застосунків.

Мета та завдання навчальної дисципліни

Метою викладання дисципліни *«Основи програмної інженерії»* є надання здобувачам освіти систематизованих знань та навичок у галузі інформаційних технологій для формування їх здатності ефективно розробляти, супроводжувати та оптимізувати програмне забезпечення, враховуючи сучасні тенденції розробки програмних продуктів

Основними завданнями вивчення дисципліни *«Основи програмної інженерії»* є:

1. Ознайомлення з основними концепціями: Надати здобувачам освіти розуміння основних теоретичних концепцій і понять програмної інженерії, таких як життєвий цикл розробки ПЗ, методології розробки, вимоги до програмного забезпечення, тестування тощо.
2. Навчання методам проектування: Розвивати навички проектування програмних систем, включаючи структурне проектування, архітектурні патерни, та інші сучасні методи.
3. Вивчення інструментів та технологій: Ознайомлювати здобувачів освіти із сучасними інструментами розробки та технологіями, необхідними для ефективної роботи програмного інженера.
4. Засвоєння навичок управління проєктами: Навчати методам та інструментам управління проєктами в галузі програмної інженерії.
5. Розвиток навичок командної роботи: Сприяти розвитку навичок ефективної комунікації та співпраці в команді при розробці програмного забезпечення.

6. Стимулювання критичного мислення: Розвивати у студентів критичне мислення та здатність аналізувати та вирішувати проблеми, що виникають під час розробки програмного забезпечення.

Загальна мета допоможе здобувачам освіти отримати необхідний набір знань та навичок для успішної кар'єри в галузі інформаційних технологій та відповідати вимогам сучасного ІТ-середовища.

Компетентності та результати навчання

У результаті вивчення навчальної дисципліни «*Основи програмування та алгоритмічні мови*» 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/storage/app/media/Fakhova%20peredvysycha%20osvita/Zatverdzeni.standarty/2021/11/30/122-Kompyuterni.nauky.30.11.pdf>) та освітньо-професійній програмі «Інженерія програмного забезпечення» (<https://devkpa.fakel.com.ua/storage/uploads/VMS0u7w5jqmS3v7eKnIdlhSlRQtVOrxE7pgajyuE.pdf>) підготовки фахових молодших бакалаврів.

Загальні компетентності:

- ЗК3.** Здатність до абстрактного мислення, аналізу та синтезу.
ЗК4. Здатність застосовувати знання в практичних ситуаціях.
ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності.
ЗК8. Здатність вчитися і оволодівати сучасними знаннями

Спеціальні (фахові, предметні) компетентності:

СК2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.

СК3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища.

СК4. Здатність здійснювати проектування та розробку програмного забезпечення.

СК8. 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.

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

2. Інформаційний обсяг навчальної дисципліни

2.1 Тематичний план

Таблиця 2.1 – Тематичний план навчальної дисципліни

№	Назва змістових модулів і тем
Блок змістових модулів 1 – Інженерні основи програмного забезпечення	
Змістовий модуль 1.1 Огляд інженерії програмного забезпечення	
Тема 1	Вступ. Основні визначення і поняття інженерії програмного забезпечення.
Тема 2	Технологія розробки програмного забезпечення
Тема 3	Інформаційні системи. Життєвий цикл програмного забезпечення
Блок змістових модулів 2 - Основи моделювання програмного забезпечення	
Змістовий модуль 2.1 Вимоги до ПЗ	
Тема 4	Суть структурного підходу до проектування
Тема 5	Вимоги до програмного забезпечення
Тема 6	Функціональні вимоги до ПЗ
Тема 7	Нефункціональні вимоги до програмного забезпечення
Тема 8	Документування функціональних та не функціональних вимог до ПЗ
Тема 9	Системні вимоги
Тема 10	Документування системних вимог.
Тема 11	Стандартизація розробки програмного забезпечення
Тема 12	Стандарти організації IEEE та CMM
Тема 13	Стандарт SPICE
Блок змістових модулів 3 - Технології розробки програмного забезпечення	
Змістовий модуль 3.1 Нотації розробки ER- діаграм	
Тема 14	Графічні нотації побудови ERD
Змістовий модуль 3.2 Методології структурного підходу	
Тема 15	Системний аналіз
Тема 16	Методологія функціонального моделювання SADT
Тема 17	Діаграма потоків даних. Нотація Йордана-Де Марко. Нотація Гейна Сарсона.
Тема 18	Сімейство методологій IDEF, принцип побудови моделей IDEF-3.
Тема 19	Забезпечення якості ПЗ
Тема 20	Методи оцінки якості тестування. Види тестів
Разом: 105 годин	

2.2 Зміст дисципліни

Блок змістових модулів 1 – Інженерні основи програмного забезпечення

Змістовий модуль 1.1 Огляд інженерії програмного забезпечення

Тема 1. Вступ. Основні визначення і поняття інженерії програмного забезпечення.

Основні поняття інженерії програмного забезпечення, поняття CASE та технологічного процесу створення програмного забезпечення. CASE-засоби, показники якісного програмного забезпечення

Тема 2. Технологія розробки програмного забезпечення

Технології розробки ПЗ, вимоги до ТРПЗ, проблеми розробки складних програм

Тема 3. Інформаційні системи. Життєвий цикл програмного забезпечення

Поняття система, інформація, інформаційна система, фази в яких перебуває інформація, компоненти інформаційної системи

Поняття життєвого циклу програмного забезпечення, модель життєвого циклу ПЗ, структура ЖЦ ПЗ

Блок змістових модулів 2 - Основи моделювання програмного забезпечення

Змістовий модуль 2.1 Вимоги до ПЗ

Тема 4. Суть структурного підходу до проектування

Визначення структурного підходу та його місце в життєвому циклі розробки ПЗ. Зв'язок структурного підходу з іншими методологіями та підходами до проектування. Основні концепції структурного підходу: Розділення програмного забезпечення на підсистеми та модулі, Визначення модульного принципу та його переваги в управлінні складністю проєктів.

Тема 5. Вимоги до програмного забезпечення

Вимоги до програмного забезпечення, розробка вимог, функціональні та не функціональні вимоги. Методи визначення вимог та їх документування. Тестування вимог: критерії та способи.

Тема 6. Функціональні вимоги до ПЗ.

Пояснення того, що таке функціональні вимоги в контексті розробки програмного забезпечення. Розрізнення між функціональними та нефункціональними вимогами. З'ясування, як функціональні вимоги визначають поведінку системи. Пояснення їх важливості для успішної розробки та відповідності вимогам замовника. Процес визначення функціональних вимог.

Тема 7. Нефункціональні вимоги до програмного забезпечення

Пояснення ролі нефункціональних вимог у створенні високоякісного програмного забезпечення. Аналіз основних категорій, таких як ефективність, надійність, безпека, доступність та інші. Зрозуміння того, як кожна категорія впливає на властивості програмного забезпечення. Огляд вимог, пов'язаних з продуктовими характеристиками, такими як

продуктивність, ефективність та масштабованість. Визначення метрик для вимірювання та оцінки цих характеристик.

Тема 8. Документування функціональних та не функціональних вимог до ПЗ.

Пояснення важливості документування вимог як ключового етапу в життєвому циклі розробки ПЗ. Визначення ролі документації у забезпеченні якості та ефективності розробки. Огляд популярних інструментів для створення специфікацій та документації функціональних вимог. Використання сценаріїв, діаграм та таблиць для наглядності та повноти опису функціональності.

Тема 9. Системні вимоги

Розгляд того, що таке системні вимоги та як вони визначають характеристики апаратної та програмної інфраструктури для ефективної роботи програмного забезпечення. Розгляд апаратних, програмних та мережевих компонентів системних вимог. Визначення їх взаємодії та впливу на продуктивність ПЗ. Визначення категорій системних вимог, таких як продуктивність, надійність, масштабованість та інші.

Тема 10. Документування системних вимог.

Визначення основних цілей документування системних вимог, таких як забезпечення однозначності, відстеження та підтримка комунікації. Опис системи. Необхідність детального опису системи, її функцій, меж та інтерфейсів. Використання діаграм, блок-схем, та інших засобів для наочного представлення структури системи. Використання текстових редакторів, CASE-інструментів та спеціалізованих платформ.

Тема 11. Стандартизація розробки програмного забезпечення

стандарт BSP (Business System Planning), Міжнародні стандарти ISO. Стандарти організації IEEE, Стандарт зрілості компанії-розробника програмного забезпечення CMM

Тема 12. Стандарти організації IEEE та CMM

Керівництво до зведення знань з програмної інженерії (SWEBOOK), Методологія CMM, ознаки зрілої організації

Тема 13. Стандарт SPICE

Визначення SPICE як стандарту, що визначає процеси оцінки та покращення програмної інженерії. Мета та користь використання SPICE. Огляд моделі процесів SPICE, яка визначає ключові аспекти управління якістю програмного забезпечення. Компоненти та зв'язки між ними.

Блок змістових модулів 3 - Технології розробки програмного забезпечення

Змістовий модуль 3.1 Нотації розробки ER- діаграм

Тема 14. Графічні нотації побудови ERD

Поглиблення в поняття та походження нотації Баркера в контексті моделювання систем та процесів. Розуміння основних цілей та областей використання нотації Баркера в інженерії програмного забезпечення. Пояснення використання структурних блоків для представлення елементів систем та їх взаємодії. Роль та значення блоків у моделюванні. Приклади

використання нотації Баркера для моделювання архітектурних рішень та компонентів програмного забезпечення.

Огляд нотації Чена як інструменту для моделювання систем та процесів у програмній інженерії. Визначення базових елементів нотації Чена, таких як блоки, стрілки та артефакти. Пояснення їхньої ролі у представленні інформації. Поглиблення в різні види діаграм Чена, включаючи діаграми структури, діаграми потоку та інші. Визначення сфер застосування кожного типу діаграми.

Змістовий модуль 3.2 Методології структурного підходу

Тема 15. Системний аналіз

Огляд поняття та ролі системного аналізу в програмній інженерії. Визначення важливості вивчення та розуміння систем для ефективної розробки програмного забезпечення. Визначення цілей системного аналізу, таких як ідентифікація проблем, моделювання процесів та визначення вимог до програмного забезпечення. Розгляд концепцій систем, їхніх елементів та взаємозв'язків. Підсумок властивостей систем, таких як емерджентність та взаємодія.

Тема 16. Методологія функціонального моделювання SADT

Системне проектування, SADT-методологія, SADT-модель, типи зв'язків між функціями, правила деталізації при побудові діаграм, поняття міні-специфікації

Побудова SADT-моделі

Виділення SADT-блоків з інтерфейсними дугами. Декомпозиція основного блоку, побудова моделі, організація діаграм в ієрархію

Тема 17. Діаграма потоків даних. Нотація Йордана-Де Марко. Нотація Гейна Сарсона

Етапи виконання проекту, контекстні діаграми нотацією Йордана-Де Марко.

Побудова початкової контекстної діаграми та матриці списку подій.

Побудова контекстної діаграми, діаграми структур даних та ERD.

Побудова діаграми потоків даних за допомогою сервису MindOnMind

Визначення DFD як графічного інструменту для моделювання потоків даних у системах. Роль DFD у визначенні та аналізі процесів та потоків даних. Визначення та розгляд процесів у DFD як блоків, що виконують конкретні функції в системі. Пояснення ролі процесів у відображенні логічної структури системи. Представлення потоків даних як стрілок, що вказують напрямок переміщення даних між процесами, об'єктами та зовнішніми сутностями. Розгляд важливості назв для коректного ідентифікування потоків.

Тема 18. Сімейство методологій IDEF, принцип побудови моделей IDEF-3

Визначення IDEF (Integrated DEFinition), сімейства методологій, розроблених для моделювання та аналізу бізнес-процесів та інших систем. Роль IDEF в інженерії програмного забезпечення та визначення його ключових принципів. Визначення мети та використання IDEF-3 в різних

областях. Визначення ключових термінів та понять, таких як об'єкти, атрибути, спостерігачі тощо. Поглиблення в структуру та взаємодію об'єктів у IDEF-3. Розгляд можливостей моделювання взаємодії між об'єктами для представлення бізнес-процесів

Тема 19. Забезпечення якості ПЗ

Огляд концепції та важливості ЗЯПЗ в процесі розробки програмного забезпечення. Роль ЗЯПЗ у забезпеченні ефективності, надійності та відповідності вимогам. Визначення основних цілей ЗЯПЗ, таких як визначення вимог, відстеження та виправлення дефектів, забезпечення відмінності функціональності. Пояснення ключових принципів ЗЯПЗ, таких як планування, валідація та верифікація. Визначення ролей у процесі ЗЯПЗ, таких як менеджер якості, тестувальник, аналітик валідації тощо. Розгляд відповідальностей кожної ролі та їх взаємодія.

Тема 20. Методи оцінки якості тестування. Види тестів

Огляд різних методів оцінки якості тестування, таких як покриття коду, аналіз дефектів, метрики продуктивності тестування. Огляд видів покриття: стрічкове, функціональне, структурне. Пояснення процесу аналізу та категоризації дефектів, виявлених під час тестування. Звернення до методів прийняття рішень на основі аналізу дефектів. Огляд функціонального тестування як методу, спрямованого на перевірку відповідності програмного забезпечення функціональним вимогам.

2.3 Перелік практичних робіт з дисципліни

Таблиця 2.2 – Тематичний план практичних робіт

№	Тема практичної роботи
3.	Розробка вимог до ІС.
4.	Розробка ТЗ.
5.	Розробка ER-діаграми за нотацією Баркера.
6.	Розробка ER-діаграми за нотацією П. Чена.
7.	Методологія функціонального моделювання SADT
8.	Діаграма потоків даних. Нотація Йордана-Де Марко.
9.	Діаграма потоків даних. Нотація Гейна Сарсона
10.	Методологія IDEF-3.
11.	Розробка тест-кейсів
	Всього: 40 годин

2.4 Кількість годин разом за програмою дисципліни

Розподіл дисципліни у годинах	
Повний обсяг часу	105
Аудиторні заняття, годин	57
із них (кількість годин):	
лекції	17
практичні	40
Самостійна робота	48

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

1. Кросплатформене програмне забезпечення для малювання графіків [сайт]/ <https://app.diagrams.net>
2. Інструмент побудови діаграм потоків даних он-лайн [сайт]/ <https://creately.com>

4. Інформаційні ресурси

Базові (основні):

1. К. М. Лавріщева Програмна інженерія: підручник. / Київ, УДК 681.3.06, 2008. – 322 с.
2. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолтНТУ, 2017. – 218 с.
3. І. Л. Бородкіна, Г. О. Бородкін Інженерія програмного забезпечення: Навчальний посібник./ Центр учбової літератури, 2020. – 204 с.

Додаткові:

1. Р. Мартін «Чиста архітектура», Фабула, 2019 – 416 с.;
2. Ю. Рамський «Проектування й опрацювання баз даних: Посібник для вчителів», Навчальна книга Богдан, 416 с;
3. Ерік Еванс «Предметно-орієнтоване проектування (DDD): структуризація складних програмних систем», Діалектика, 2016 – 448 с..
4. Ю. Грицюк «Аналіз вимог до програмного забезпечення», Львівська Політехніка, 2018 – 456 с.
5. О. Перевозчикова «Інформаційні системи і структури даних», КиєвоМогилянська академія, 2007 – 288с

5. Форма підсумкового контролю

Екзамен (VI семестр), поточний контроль.

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. Основні поняття:
 - а. Що таке програмна інженерія?
 - б. Різниця між програмним забезпеченням та програмною інженерією.
 - с. Поясніть терміни "процес розробки програмного забезпечення" та "життєвий цикл програмного продукту".
2. Моделі життєвого циклу програмного забезпечення:
 - а. Які основні моделі життєвого циклу програмного забезпечення ви знаєте?
 - б. Переваги та недоліки кожної моделі.
3. Вимоги та аналіз:
 - а. Як здійснюється аналіз вимог до програмного продукту?
 - б. Що таке функціональні та нефункціональні вимоги?
4. Проектування:
 - а. Які основні етапи проектування програмного забезпечення?
 - б. Різниця між архітектурним та детальним проектуванням.
5. Реалізація та тестування:
 - а. Що включає в себе етап реалізації програмного забезпечення?
 - б. Основні види тестування програм.
6. Керування проектом:
 - а. Які інструменти використовуються для керування проектом в програмній інженерії?
 - б. Як визначити обсяг робіт та встановити терміни виконання?
7. Якість та тестування якості:
 - а. Що означає термін "якість програмного забезпечення"?
 - б. Як проводиться тестування якості?
8. Конфігураційне управління:
 - а. Що таке конфігураційне управління в програмній інженерії?
 - б. Які інструменти використовуються для керування конфігурацією програмного забезпечення?
9. Інженерія вимог:
 - а. Як розробляти вимоги до програмного продукту?
 - б. Що таке принцип "однозначності" в інженерії вимог?

10. Етика та професійна відповідальність:

- а. Які питання етики можуть виникнути в галузі програмної інженерії?
- б. Як можна враховувати етичні аспекти при розробці програмного забезпечення?