

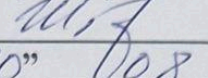
ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з методичної роботи

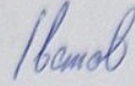
 Ярослав ШТАНЬКО
"30" 08 2024 р.

ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	перший (бакалаврський)
Статус дисципліни:	обов'язкова
Мова викладання, навчання та оцінювання:	українська

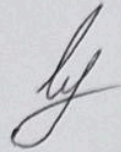
РОЗРОБНИК:
PhD, доцент кафедри ІТ



Олександр ІВАНОВ

ЗАТВЕРДЖЕНО:

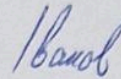
на засіданні кафедри ІТ
протокол № 1 від 28.08 2024 р.
К. т. н., завідувач кафедри



Сергій ВАЩИШАК

УЗГОДЖЕНО:

Гарант ОПП



Олександр ІВАНОВ

СХВАЛЕНО:

на засіданні Науково-методичної ради, протокол № 1 від 30.08 2024 р.

e-mail	oleksandr.o.ivanov@ukd.edu.ua
Номер аудиторії чи кафедри	Кафедра ІТ
Посилання на сайт	Олександр Іванов
Сторінка курсу в СДО	<u>Інженерія програмного забезпечення</u>

ВСТУП

Анотація навчальної дисципліни «Інженерія програмного забезпечення»

Програмне забезпечення, в різних його проявах, є невід'ємною частиною сучасного світу, яка не тільки полегшує роботу та відкриває нові можливості, але, в певному роді, і формує наше майбутнє. Програмне забезпечення використовується у всіх галузях людської діяльності – виробництво, обслуговування, послуги, розваги. Якщо задуматись, кожен з нас зможе згадати і назвати приклади того, як певні «програми» в прямому сенсі змінювали світ. Можливо, прямо зараз Ви очікуєте оновлення «додатку», яким користуєтесь повсякденно, оновлення, що доповнить його новими функціями, можливостями, зміною інтерфейсу або просто виправить певні недоліки. З іншого боку, можна згадати приклади, коли оновлення призвели до різкого зниження популярності «програми» або ж його повного краху на динамічному ринку.

В переважній більшості випадків, навіть саме «просте» програмне забезпечення, яким ми користуємось повсякденно, проходить складний та тривалий шлях, який, окрім безпосередньої розробки, включає визначення цілей, аудиторії, аналіз аналогів, тестування і багато іншого. В сучасному цифровому світі часто недостатньо, щоб програмне забезпечення лише виконувало поставлені задачі, кожен елемент відіграє значну роль, від першочергового плану та завдань до контролю якості та дизайнерських рішень.

В даному курсі буде розвинуто та практично застосовано абстрактне мислення, аналіз і синтез для проектування, моделювання і опису програмного забезпечення, з урахуванням зацікавленої аудиторії та завдань, на вирішення яких направлений продукт. Також, буде розглянуто історичні аспекти популярних продуктів, їх найбільш визначні та ключові моменти, які змінили уявлення про програмне забезпечення та цифровий світ в цілому, для визначення найважливіших факторів, що безпосередньо впливали і впливають на ринок програмних продуктів. Курс направлений на вміння працювати командно та самостійно над програмним продуктом, включаючи всі етапи його «життєвого циклу» та підтримки для забезпечення ефективної роботи і популярності.

Мета та завдання навчальної дисципліни: навчитися не тільки безпосередньо створювати якісне програмне забезпечення, але і

аналізувати існуючі продукти, виокремлювати основні елементи, що забезпечили популярність або навпаки – зниження популярності застосування; цілі та завдання, які повинен виконувати продукт, аудиторію користувачів; можливості реалізації та забезпечення універсальності використання враховуючи сучасні тенденції, стилістичні та дизайнерські рішення та зацікавлення потенційних користувачів.

В результаті вивчення дисципліни студент повинен **знати**:

- призначення та види проектів;
- основні етапи та кроки створення програмного продукту;
- методики, поняття і підходи проектування, розробки, тестування та підтримки програмного забезпечення;
- підходи до реалізації програмного продукту;
- ключові фактори та задачі, що можуть виникнути і (або) впливають на програмне забезпечення на різних етапах розробки і тестування;
- задачі, вимоги та обов'язки спеціалістів з інженерії програмного забезпечення.

В результаті вивчення дисципліни студент повинен **вміти**:

- застосовувати знання і практичні навички з інженерії програмного забезпечення (та суміжних галузей) для аналізу і постановки завдань щодо розробки та тестування програмного забезпечення;
- вибирати та практично застосовувати методи і підходи створення програмного забезпечення;
- планувати та реалізовувати проекти в галузі інженерії програмного забезпечення командно та самостійно;
- ефективно реалізовувати розробку, тестування та контроль якості програмного забезпечення.

Компетентності та результати навчання, яких набувають здобувачі освіти внаслідок вивчення навчальної дисципліни (шифри та зміст компетентностей та програмних результатів навчання вказано відповідно до ОПІ “Розробка та тестування програмного забезпечення”.

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
<p>ІК. Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.</p>	<p>ПРН5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктноорієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p>
<p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p>	<p>ПРН6. Уміння вибирати та використовувати відповідні задачі методологію створення програмного забезпечення.</p>
<p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p>	<p>ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p>
<p>ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p>	
<p>ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p>	
<p>ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p>	
<p>ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p>	
<p>ФК14. Здатність до алгоритмічного та логічного мислення.</p>	

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	2		
Семестр	3		
Кількість кредитів ЄКТС	6		
Аудиторні навчальні заняття		денна форма	заочна форма
	лекції	32 (в годинах)	6 (в годинах)
	семінари, практичні	28 (в годинах)	6 (в годинах)
Самостійна робота		120 (в годинах)	168 (в годинах)
Форма підсумкового контролю	екзамен		

Структурно-логічна схема вивчення навчальної дисципліни:

Пререквізити	Постреквізити
Основи програмування	Конструювання програмного забезпечення
Математичний аналіз	Об'єктно-орієнтоване програмування
Комп'ютерна дискретна математика	Менеджмент проектів програмного забезпечення
Інформаційно-комунікаційні технології	UI/UX дизайн
Основи роботи з нейронними мережами	Документування програмних продуктів

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Перелік тем лекційного матеріалу

Тема 1. Інженерія програмного забезпечення в сучасному світі (2 год)

Як інженерія програмного забезпечення формує сучасний цифровий світ. Життєвий цикл програмного забезпечення. Основні етапи життєвого циклу програмного забезпечення. Фактори, що впливають на програмний продукт. Цілі, завдання та структура курсу.

Самостійне вивчення (4 год): обрати популярний сьогодні (або в минулому) програмний продукт та ознайомитись з історією його створення [1, 3, 4].

Тема 2. Керування проектами (4 год)

Суть керування проектами. Структури проектів. Завдання проект-менеджменту. Визначення цілей, задач та потенційно зацікавлених користувачів. Поділ на команди та розподіл обов'язків серед учасників команди. Попередня оцінка проекту. Прототипування. Основні етапи проекту програмного забезпечення.

Самостійне вивчення (5 год): знайти та проаналізувати роботу проект-менеджера на основі існуючих проектів чи програмного забезпечення [2, 4].

Тема 3. Життєвий цикл програмного забезпечення (4 год)

Методологія і послідовність створення та підтримки якісного програмного продукту. Вимоги до підходів планування, створення і підтримки продукту. Аналіз вимог, планування, проектування, розробка, тестування, впровадження та підтримка. Фактори, ризики, явища та ситуації, що впливають програмний продукт на різних стадіях.

Самостійне вивчення (5 год): для реально існуючого популярного сьогодні (або в минулому) програмного продукту проаналізувати життєвий цикл, основні етапи існування, причини піків популярності (або зниження популярності), теоретичні можливості до застосування такого аналізу в майбутньому для планованого продукту чи проекту [1, 4].

Тема 4. Проектування архітектури програмного забезпечення (6 год)

Задачі проектування архітектури програмного забезпечення. Патерни проектування. Високорівневі патерни. Принципи опису та документування

архітектури. Діаграми для опису та документування архітектури. Види архітектур. Метрики для оцінки етапів проектування архітектури.

Самостійне вивчення (5 год): визначити переваги та недоліки у патернах проектування [1]. Ознайомитись з архітектурою Restful та проектуванням програмного забезпечення з урахуванням такої архітектури.

Тема 5. Реалізація програмного забезпечення (4 год)

Стратегія реалізації програмного забезпечення. Засоби реалізації. Реалізація компонентів. Основні стандарти та методики написання коду. «Простота» та «зрозумілість» програмного коду, види кодування. Системи контролю версій. Робота з Git.

Самостійне вивчення (6 год): ознайомитись з реальними стратегіями реалізації програмного забезпечення, виокремити основні елементи стратегії і як це впливало на успіх (чи невдачі) проекту чи програмного забезпечення. [2]

Ознайомитись з процедурою дебагу, інструментами для дебагу та вирішенням конфліктів в системах контролю версій [3].

Тема 6. Тестування програмного забезпечення (6 год)

Види і методики тестування. Побудова стратегії тестування та тест-плану. Закрите та відкрите тестування. Особливості тестування програмного забезпечення. Інтеграції. Аналіз та забезпечення якості і надійності програмного забезпечення. Кросплатформенність.

Самостійне вивчення (5 год): види та особливості проведення тестування; причини, переваги застосування різних видів тестування; інструменти проведення тестування програмного забезпечення [2, 3].

Тема 7. Розгортання програмного забезпечення (4 год)

Розгортання та встановлення програмного забезпечення. Супровідна документація. Реалізація програмного продукту на різних платформах.

Самостійне вивчення (4 год): ознайомитись з проведенням автоматизації, систематизації та прискорення процесу доставки змін в кодї до кінцевого споживача за допомогою Continuous Integration (CI) та Continuous Deployment (CD); принципи застосування та проведення [1].

Тема 8. Пост-релізна підтримка програмного забезпечення (2 год)

Підтримка програмного забезпечення. Дорожня карта продукту. Аналіз та оновлення програмного забезпечення з моменту реалізації.

Самостійне вивчення (6 год): проаналізувати приклади реальних успішних та невдалих прикладів пост-релізної підтримки; важливість застосування дорожніх карт проектів [4].

Зміст практичних занять

Тема 1. Програмні продукти в цифровому світі (2 год)

Особливості першого програмного забезпечення. Відмінності і призначення різних видів програмного забезпечення. Причини успіху та невдач програмного забезпечення. Фактори, що впливають на «еволюцію» і створення нових програмних продуктів.

Аналіз історії розвитку конкретного програмного забезпечення, що включає його проектування, реалізацію, підтримку та сучасний стан.

Самостійна робота (5 год): на основі реального проекту або програмного забезпечення, для розробки конкурентоспроможного аналогу скласти план роботи проект-менеджера, визначити цілі, задачі, потенційно зацікавлених учасників [18, 19].

Тема 2. Ініціація проекту (2 год)

Ініціація проекту як перший етап життєвого циклу управління проектом. Визначення бізнес-проблеми і (або) можливостей, формування проекту, визначення та розподіл обов'язків команди. Розгляд та детальний опис проблем і (або) можливостей, альтернативних рішень, переваг та недоліків. Формування резюме проекту.

Самостійна робота (5 год): на прикладі існуючого або запланованого програмного продукту визначити бізнес-проблему і (або) можливість, описати альтернативні рішення, проаналізувати переваги, недоліки та вимоги [4, 18, 19]. Спробувати провести поглиблену ініціацію проекту, що включає детальний аналіз, залучення зацікавлених сторін, визначення обсягу та аудиторії, розгляд ризиків, витрат та очікуваної користі. Сформувати резюме проекту [18, 19].

Тема 3. Формування технічного завдання проекту (2 год)

Технічне завдання проекту. Підходи та основні принципи формування технічного завдання. Історії користувача (User Stories). Використання

історій користувача для формування технічного завдання проекту. Критерії INVEST (Independent Negotiable Valuable Estimable Small Testable).

Самостійна робота (5 год): описати завдання проекту (конкретного програмного продукту) та сформулювати технічне завдання для команди розробників на основі історій користувача [18, 19].

Тема 4. UML-діаграми та матриці відповідностей (2 год)

Важливість уніфікованої мови моделювання (Unified Modeling Language) в проектах. Діаграма розгортання. Зв'язок частин програмного забезпечення з пристроями, на яких вони будуть виконуватись. Застосування програм для виконання UML-діаграми. Візуалізація взаємодії програмного забезпечення з апаратним забезпеченням. Визначення ролей та учасників проекту. Використання матриці відповідності (Responsibility Assignment Matrix) для розподілу ролей та обов'язків в рамках проектів.

Самостійна робота (5 год): користуючись програмним забезпеченням виконати UML-діаграму розгортання існуючого або планованого проекту; побудувати матрицю відповідності для планованого проекту [2].

Тема 5. Інформаційна модель керування даними проекту (4 год)

Інформаційна модель керування даними проекту як інструмент для розробки та управління. Моделювання даних та використання ERD (Entity Relationship Diagram). Створення ERD. Концептуальні, логічні та фізичні моделі.

Самостійна робота (5 год): ознайомитись з особливостями моделювання даних і використання ERD на прикладі існуючих проектів. Скласти ERD для запланованого програмного забезпечення [2].

Тема 6. Діаграма варіантів використання (4 год)

Застосування діаграми варіантів використання в UML. Визначення контексту системи, уточнення функціональної поведінки і визначення документація, що може застосовуватись для обговорення системи між розробниками, замовниками та іншими зацікавленими особами. Документація для розробки.

Самостійна робота (5 год): розглянути та проаналізувати конкретні приклади використання діаграми варіантів використання [2].

Тема 7. Діаграми класів та діаграми компонентів (4 год)

Застосування діаграм класів та діаграм компонентів в UML. Класи, атрибути, методи та їх взаємодія. Побудова коду для розробки програмного забезпечення. Моделювання великих систем у менші підсистеми. Переваги та особливості застосувань діаграм в конкретних випадках. Створення діаграм для опису проєктованого програмного забезпечення.

Самостійна робота (5 год): навести аргументи щодо важливості застосування діаграм класів та діаграм компонентів в UML; ознайомитись з існуючими прикладами; скласти діаграми для запланованого програмного забезпечення [2].

Тема 8. Графічний інтерфейс користувача (4 год)

Взаємодія користувача з електронними пристроями через графічний інтерфейс користувача (Graphical User Interface – GUI). Особливості GUI на різних пристроях та системах. Важливі аспекти GUI. Дизайн інтерфейсу. Діаграма станів (State Diagram). Прототипування інтерфейсів.

Самостійна робота (5 год): проаналізувати GUI реального програмного забезпечення, визначити переваги, особливості та недоліки; використовуючи програми, на основі GUI реального програмного забезпечення, створити прототип візуального представлення елементів інтерфейсу, розташування елементів, їх розміри та взаємодію [18, 19].

Тема 9. Планування методології розробки програмного забезпечення (2 год)

Види методологій розробки програмного забезпечення. Особливості, переваги та недоліки методологій. Методології Waterfall, Agile, Scrum. Вибір методології реалізації проєкту. Інтерактивний підхід до розробки програмного забезпечення.

Самостійна робота (5 год): на основі реального програмного забезпечення проаналізувати методологію розробки; запропонувати методологію для планованого програмного забезпечення [2, 3].

Тема 10. Тест-плани та тестові стратегії (2 год)

Особливості створення та застосування тест-планів та тестових стратегій. Тестування програмного забезпечення. Пост-релізна підтримка програмного забезпечення на основі взаємодії учасників проєкту.

Самостійна робота (5 год): проаналізувати види тестування програмного забезпечення, навести приклади в яких випадках доцільно застосовувати окремі види [3].

Проаналізувати випадки та ситуації, коли пост-релізна підтримка кардинально змінювала ситуацію, пов'язану з існуючими проектами чи програмним забезпеченням [2].

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни:

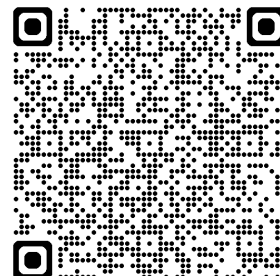
Найменування видів робіт	Розподіл годин за формами навчання	
	денна	заочна
Самостійна робота, год, у т.ч.:	120	168
Опрацювання матеріалу, викладеного на лекціях	20	20
Підготовка до практичних занять та контрольних заходів	30	20
Підготовка звітів з практичних робіт	20	30
Підготовка до поточного контролю	10	30
Опрацювання матеріалу, винесеного на самостійне вивчення	40	68

ПОЛІТИКА КУРСУ

Коротко, з покликанням на відповідну нормативну базу УКД, висвітлити питання.¹

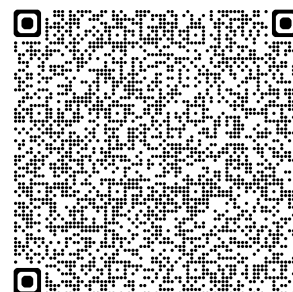
1) щодо системи поточного і підсумкового контролю

Організація поточного та підсумкового семестрового контролю знань студентів, проведення практик та атестації, переведення показників академічної успішності за 100-бальною шкалою в систему оцінок за національною шкалою здійснюється згідно з “Положенням про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти”. Ознайомитись з документом можна за покликанням.



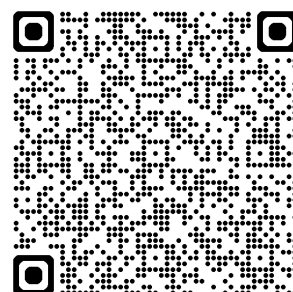
2) щодо оскарження результатів контрольних заходів

Здобувачі вищої освіти мають право на оскарження оцінки з дисципліни отриманої під час контрольних заходів. Апеляція здійснюється відповідно до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за покликанням.



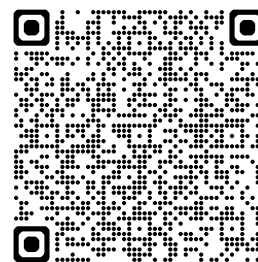
3) щодо відпрацювання пропущених занять

*Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з **конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку)**, якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за покликанням.*



4) щодо дотримання академічної доброчесності

“Положення про академічну доброчесність” закріплює моральні принципи, норми та правила етичної поведінки, позитивного, сприятливого, доброчесного освітнього і наукового середовища, професійної діяльності та професійного спілкування спільноти Університету, викладання та провадження наукової (творчої)

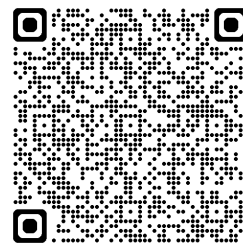


¹ зміст пунктів може редагуватись з огляду на особливості курсу

діяльності з метою забезпечення довіри до результатів навчання. Ознайомитись з документом можна за [покликанням](#).

5) щодо використання штучного інтелекту

“Положення про академічну доброчесність” визначає політику щодо використання технічних засобів на основі штучного інтелекту в освітньому процесі. Ознайомитись з документом можна за [покликанням](#).² “Положення про систему запобігання та виявлення академічного плагіату, самоплагіату, фабрикації та фальсифікації академічних творів” містить рекомендації щодо використання в академічних текстах генераторів на основі штучного інтелекту. Ознайомитись з документом можна за [покликанням](#).



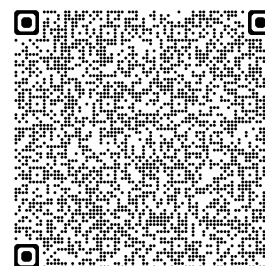
6) щодо використання технічних засобів в аудиторії та правила комунікації

Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо, окрім виробничої необхідності. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання.

Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

7) щодо зарахування результатів навчання, здобутих шляхом формальної/інформальної освіти

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [покликанням](#).³



Окремі теми можуть бути зараховані за рахунок публікації одноосібних або у співавторстві статей, тез, виступів на конференціях. Можливість зарахування попередньо узгоджується з викладачем з позицій актуальності теми, журналу, конференції, тощо.

Також, студентам надається можливість перерахування неформальної освіти за рахунок проходження з отриманням сертифікату сертифікованих курсів на освітніх платформах. Можливість зарахування попередньо узгоджується з викладачем з позицій актуальності курсу, сертифікату, тривалості проходження, тощо.

МЕТОДИ НАВЧАННЯ

² визначається політика використання ІІІ в навчальній дисципліні - дозволене/заборонене, правила використання

³ визначається перелік електронних та інших ресурсів та умови перерахування

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця.

Програмний результат навчання⁴	<u>Метод навчання</u>	Метод оцінювання
ПРН2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	словесні методи (лекція, розповідь-пояснення) наочні методи (комп'ютерні і мультимедійні методи) інтерактивні методи (кейс-метод, дискусія) рольові і ділові ігри	поточний контроль усний контроль
ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.	словесні методи (лекція, розповідь-пояснення) наочні методи (комп'ютерні і мультимедійні методи) практичні методи (практичні роботи) індуктивний дедуктивний	поточний контроль усний контроль тестовий контроль підсумковий, іспит
ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	наочні методи (комп'ютерні і мультимедійні методи) практичні методи (практичні роботи) індуктивний дедуктивний проблемно-пошуковий	поточний контроль тестовий контроль підсумковий, іспит
ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.	наочні методи (комп'ютерні і мультимедійні методи) практичні методи (практичні роботи) індуктивний дедуктивний проблемно-пошуковий	поточний контроль тестовий контроль підсумковий, іспит

⁴ для вибіркового навчання дисциплін вказується результат навчання

ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.	словесні методи (лекція, розповідь-пояснення) наочні методи (комп'ютерні і мультимедійні методи) практичні методи (практичні роботи) індуктивний дедуктивний	поточний контроль усний контроль тестовий контроль
---	--	--

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Вид	Зміст	% від загальної оцінки	Бал	
			min	max
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	екзамен	40	24	40
Всього:	-	100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація **поточного** контролю здійснюється в “Електронному журналі обліку успішності академічної групи” на підставі чотирибальної шкали – “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».⁵

⁵ можна вказати теми чи завдання, які є обов'язковими до виконання, а також особисті підходи до оцінювання рівня знань здобувачів під час аудиторної роботи

Критерії оцінювання:

«незадовільно»	володіє навчальним матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів, що позначаються окремими словами чи реченнями; володіє матеріалом на елементарному рівні засвоєння, викладає його уривчастими реченнями, виявляє здатність висловити думку на елементарному рівні; володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу; не може розпізнати або відтворити матеріал практичних завдань
«задовільно»	володіє матеріалом на початковому рівні, значну частину матеріалу відтворює на репродуктивному рівні; володіє матеріалом на рівні, вищому за початковий, здатний за допомогою викладача логічно відтворити значну його частину; може відтворити значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, за допомогою викладача може аналізувати навчальний матеріал, порівнювати та робити висновки, виправляти допущені помилки; може розпізнати або відтворити за прикладом матеріал практичних занять
«добре»	здатний застосовувати вивчений матеріал на рівні стандартних ситуацій, частково контролювати власні навчальні дії, наводити окремі власні приклади на підтвердження певних тверджень: вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати її на практиці, контролювати власну діяльність, виправляти помилки і добирати аргументи на підтвердження певних думок під керівництвом викладача; вільно володіє вивченим обсягом матеріалу, та вміє застосовувати його на практиці; вільно розв'язує задачі в стандартних ситуаціях, самостійно виправляє допущені помилки, добирає переконливі аргументи на підтвердження вивченого матеріалу;
«відмінно»	виявляє початкові творчі здібності, самостійно визначає окремі цілі власної навчальної діяльності, оцінює окремі нові факти, явища, ідеї; знаходить джерела інформації та самостійно використовує їх відповідно до цілей, поставлених викладачем; здатний вільно дискутувати на теми, пов'язані з матеріалом навчальної дисципліни, висловлювати власні думки, визначати програму особистої діяльності, самостійно оцінює різноманітні

	явища і факти, виявляючи особисту позицію щодо них; без допомоги викладача знаходить джерела інформації і використовує одержані відомості відповідно до мети та завдань власної пізнавальної діяльності; використовує набуті знання і вміння в нестандартних ситуаціях; виявляє вміння знаходити альтернативні шляхи для вирішення завдань, вміє самостійно здобувати знання.
--	---

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення дисципліни впродовж наступного навчального семестру. За результатами підсумкового контролю (диференційований залік/екзамен) студент може отримати 40 балів. Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен/диференційований залік і повинні йти на перездачу. Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно з **Шкалою оцінювання знань за ЄКТС**) і є сумою балів отриманих під час поточного та підсумкового контролю.

Шкала оцінювання знань за ЄКТС:

Оцінка за національною шкалою	Рівень досягнень, %	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-

Не зараховано	0 – 59	-
---------------	--------	---

Студенти, які не з'явилися на екзамені без поважних причин, вважаються такими, що одержали незадовільну оцінку.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ⁶

Основна література

1. Баран. С. Розробка програмного забезпечення з використанням патернів проектування: Навчальний посібник. Кривий Ріг: ДУЕТ, 2023. 203 с.
2. Трофименко О., Манаков С., Ларін Д. Основи програмної інженерії: навчально-методичний посібник. Одеса: Фенікс, 2022. 194 с.
3. Крепич С., Співак І. Якість програмного забезпечення: базовий курс. Тернопіль: ФОП Паляниця В.А., 2020, 478 с.
4. Бабенко М. Професійна практика програмної інженерії: конспект лекцій. Кам'янське: ДДТУ, 2021. 88 с.

Додаткова література

5. Смагіна О. Якість програмного забезпечення та тестування: навчальний посібник. Старобілівськ: ДЗ «ЛНУ ім. Тараса Шевченка», 2021. 286 с.
6. Бородкіна І., Бородкін Г. Інженерія програмного забезпечення: посібник для студентів вищих навчальних закладів. Київ: НУБПУ, 2018. 251 с.
7. Sardar Mudassar Khan. Software Requirements Engineering Practical Approach. C# Corner India, 2024. 189 p.
8. Rehman M., Amin A., Gilal A., Hashmani M. Human Factors in Global Software Engineering. IGI Global, 2019. 381 p.

Електронні інформаційні ресурси

9. Стрюк, А. М. Становлення та розвиток інженерії програмного забезпечення як галузі знань. Інформаційні технології в освіті. 2018. № 37 (4). С. 103-136. URL: <http://ekhsuir.kspu.edu/123456789/17585>
10. Стрюк, А.М. Інженерія програмного забезпечення: перші 50 років становлення та розвитку. CEUR Workshop Proceedings, 2018. 2292, pp. 11-36 URL: <https://ceur-ws.org/Vol-2292/paper01.pdf>

⁶ обов'язково: враховувати вимоги [ДСТУ 8302:2015](#) (відповідно до [Наказу № 65, від 4.03.2016](#)), [рекомендації](#) Національного агентства з забезпечення якості вищої освіти, використовувати літературу за останні 5-7 років, наводити власні публікації за змістом навчальної дисципліни.

11. Altenternative. Комп'ютерні науки або Інженерія програмного забезпечення? В чому різниця? URL:
<https://www.youtube.com/watch?v=l5PSODYkWbc>
12. #Software #Engineering Full Course. URL:
https://www.youtube.com/playlist?list=PL_pbwdlyffslgxMVyXhnHiSn_EWTvx1G-
13. What Do Software Engineers ACTUALLY Do? URL:
<https://youtu.be/iIxZrYzJJ7I?si=v3FruUW0XTr4ZhOj>
14. Wesley Chai, Rachel Lebeaux. What is an IT project manager? URL:
<https://www.techtarget.com/searchcio/definition/IT-project-manager#:~:text=An%20IT%20project%20manager%20is,organizations%20rely%20on%20computing%20technologies.>
15. The Fullstack Academy Team. What Do Software Engineers Do? Job Types, Training, and Salary. 2024. URL:
<https://www.fullstackacademy.com/blog/what-do-software-engineers-do>
16. Quora. What is the greatest software ever produced? URL:
<https://www.quora.com/What-is-the-greatest-software-ever-produced>
17. Nanbox. A Comprehensive History of Software Development: From Beginnings to Present. 2023. URL:
https://nanbox.com/the-history-of-software-development-over-70-years-of-innovation/#Final_Words
18. Noel Braganza. The most popular software & technical tools users picked on MuchSkills. 2024. URL:
<https://www.muchskills.com/blog/top-software-technical-tools-muchskills>
19. Srujana Maddula. Most Used Software: Top 17 Tools You Have To Know. 2024. URL: <https://mspoweruser.com/most-used-software/>
20. Institute of Data. Understanding the Role of a Video Game Software Engineer. URL:
<https://www.institutedata.com/blog/role-of-video-game-software-engineer/>