



МЕТОДОЛОГІЯ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (SOFTWARE ENGINEERING METHODOLOGY)

Робоча програма навчальної дисципліни (Силабус)

• Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F2 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Професійної та практичної підготовки (нормативна)</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, 2 семестр</i>
Обсяг дисципліни	<i>4 кредити, 120 годин, з яких 46 години аудиторних (30 год лекції, 16 год лабораторні роботи, (74 годин становить самостійна робота)</i>
Семестровий контроль/ контрольні заходи	<i>Усний залік, МКР</i>
Розклад занять	<i>http://rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент кафедри ІПЗЕ, Залевська Ольга Валеріївна o.zalevska@kpi.ua¹ Практичні роботи: к.т.н., доцент кафедри ІПЗЕ, Залевська Ольга Валеріївна o.zalevska@kpi.ua</i>
Розміщення курсу	<i>Платформа дистанційного навчання Сікорський, Telegram та E-mail. Викладені матеріали: Лабораторні роботи Домашні завдання, Література.</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Силабус навчальної дисципліни «Методологія інженерії програмного забезпечення (Software engineering methodology)» складено відповідно до освітньо-професійної програми «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» другого рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення».

Метою навчальної дисципліни є формування та закріплення у здобувачів освіти наступних компетентностей: здатність до абстрактного мислення (ЗК 01), Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК 01), Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів (ФК 03), Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення (ФК 05), аналізу та синтезу здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами у сфері інженерії програмного забезпечення (ФК 06), Здатність критично осмислювати проблеми у

¹Електронна пошта викладача або інші контакти для зворотного зв'язку, можливо зазначити прийомні години або години для комунікації у разі зазначення контактних телефонів. Для силабусу дисципліни, яку викладає багато викладачів (наприклад, історія, філософія тощо) можна зазначити сторінку сайту де представлено контактну інформацію викладачів для відповідних груп, факультетів, інститутів.

галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах (ФК 07), Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення (ФК 08), Здатність забезпечувати якість програмного забезпечення (ФК 09), Здатність проєктувати та розробляти розподілені програмні системи на основі сучасної методології інженерії програмного забезпечення(ФК 12).

Предметом вивчення навчальної дисципліни є систематичне вивчення, організація та використання методів, технік, інструментів та процесів для ефективної розробки високоякісного програмного забезпечення. Це включає в себе: принципи та практики розробки програмного забезпечення, методології управління проектами в сфері ІТ, процеси життєвого циклу програмного забезпечення, техніки оцінки та покращення якості програмного продукту

Стандарти та норми, що регламентують процес розробки ПЗ. є сучасні моделі, методи та технології професійного управління проектами для розвитку ІТ-індустрії, які базуються на філософії життєвих циклів проєктів та програм, орієнтованих на кінцеві результати під час вирішення професійних та наукових задач.

Програмні результати навчання, на формування та покращення яких спрямована дисципліна: знати і застосовувати сучасні професійні стандарти і інші нормативно- правові документи з інженерії програмного забезпечення (ПРН 01), оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу(ПРН 02), будувати і досліджувати моделі інформаційних процесів у прикладній області (ПРН 03), виявляти інформаційні потреби і класифікувати дані для проєктування програмного забезпечення (ПРН 04), розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення (ПРН 05), розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів (ПРН 06), аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення (ПРН 07), розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника (ПРН 08), обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення (ПРН 09), модифікувати існуючі та розробляти нові алгоритмічні рішення детального проєктування програмного забезпечення (ПРН 10), забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики (ПРН 11), конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу (ПРН 13), прогнозувати розвиток програмних систем та інформаційних технологій (ПРН 14), здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника (ПРН 15), планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення (ПРН 16), збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела (ПРН 17), Вміти проєктувати та розробляти розподілені програмні системи на основі сучасної методології інженерії програмного забезпечення (ПРН 20).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

У структурно-логічній схемі навчання дисципліна «Методологія інженерії програмного забезпечення / Software Engineering Methodology» розміщена тоді, коли студенти вже прослухали навчальні дисципліни з «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці

/ Development of Internet of Things and Sensor Networks Applications in Energy Industry» (ПО 05), «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці / Development of Internet of Things and Sensor Networks Applications in Energy Industry (ПО 06), «Інтелектуальний аналіз даних для задач енергетики / Intelligent Data Analysis for Energy Industry Tasks» (ПО 07) та «Хмарні та Грід-технології / Cloud and GRID Technologies» (ПО 09), що достатньо для виконання практичних робіт з даної дисципліни.

Дисципліна «Методологія інженерії програмного забезпечення / Software Engineering Methodology» забезпечує підготовку до «Практики / Practice» (ПО 02) та «Виконання магістерської дисертації / Execution of Master's Thesis» (ПО 03), які викладаються у наступних семестрах.

3. Зміст навчальної дисципліни

РОЗДІЛ 1. МОДЕЛІ ТА ФАЗИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Тема 1.1. Вступ

Тема 1.2. Визначення моделей програмної інженерії.

Тема 1.2. Моделі програмної інженерії

Тема 1.3. Етапи розробки програмного забезпечення (вимоги, проектування, реалізація, тестування).

Тема 1.4. Ролі та відповідальності на кожній фазі.

РОЗДІЛ 2. УСТАНОВКА ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тема 2.1. Процес установки програмного забезпечення.

Тема 2.2. Методи супроводу програмного забезпечення під час установки.

Тема 2.3. Супровід програмного забезпечення після установки

РОЗДІЛ 3. МЕТОДИ ТА ТЕХНІКИ, ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Тема 3.1. Методи об'єктно-орієнтованої інженерії

Тема 3.2. Методи функціонального програмування.

Тема 3.3. Техніки TDD, BDD, рефакторинг

Тема 3.4. Інструменти програмної інженерії та системи контролю версій

Тема 3.5. Мікросервісні технології програмної інженерії та хмарні обчислення

РОЗДІЛ 4. УПРАВЛІННЯ ПРОГРАМНИМ ПРОЄКТОМ. ГРУПОВА РОЗРОБКА ПРОЄКТІВ ТА КЕЙСІВ

Тема 4.1. Планування та оцінка проєкту. Розподіл ресурсів, контроль якості та управління ризиками

Тема 4.2. Використання методологій Scrum та Kanban

Тема 4.3. Управління конфліктами

РОЗДІЛ 5. ЕТИЧНІ АСПЕКТИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Тема 5.1. Відповідальність розробників ПЗ, законодавча та нормативна.

Тема 5.2. Приватність та безпека даних

Тема 5.3. Відповідальне використання штучного інтелекту

РОЗДІЛ 6. НОВІТНІ ТЕНДЕНЦІЇ В ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тема 6.1. Штучний інтелект та машинне навчання

Тема 6.2. Поняття DevOps, CI/CD.

Тема 6.3. Технології віртуальної та доповненої реальності

4. Навчальні матеріали та ресурси

Основна література

1. Winters Titus, Tom Maneesh, Hires Kunja "Software Engineering at Google: Lessons Learned from Programming Over Time." O'Reilly Media, 2020.

2. Лаврищева, К.М. "Методологія програмної інженерії." Київ: Академперіодика, 2006

3. Shari Lawrence Pfleeger "Software Engineering: Theory and Practice" авторства Сарі Пфлігер, 4-е видання, Pearson, 2009

4. Павловська, Т.А. "Методологія створення програмних систем." Київ: Видавничий дім "Академперіодика", 2002.

5. "Software Engineering Processes: Principles and Applications" by Rajib Mall, CRC Press, 2018
6. "Agile Software Development: Principles, Patterns, and Practices" by Robert C. Martin, Prentice Hall, 2017
7. "Software Engineering for Absolute Beginners" by Laurence Svekis, Apress, 2020.

Додаткова література:

1. "Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies" by Capers Jones and Jitendra Singh, McGraw-Hill Education, 2020.
2. "Refactoring: Improving the Design of Existing Code" by Martin Fowler, Addison-Wesley Professional, 2018.
3. "Domain-Driven Design: Tackling Complexity in the Heart of Software" by Eric Evans, Addison-Wesley Professional, 2003 (перевидання 2018 року).
4. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley, Addison-Wesley Professional, 2010.
5. "The Pragmatic Programmer: From Journeyman to Master" by Andrew Hunt and David Thomas, Addison-Wesley Professional, 2019 (20th Anniversary Edition).

Навчальний контент

5. Методика опанування навчальної дисципліни(освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
РОЗДІЛ 1 МОДЕЛІ ТА ФАЗИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ	
1	Тема 1.1. Вступ до предмету Мета, об'єкт, предмет та задачі курсу. Рейтингова система оцінювання слухачів курсу. Тема 1.2 Визначення моделей програмної інженерії. Моделі життєвого циклу. Водоспадна модель. Модель прототипування. Модель великого вибуху. V-подібна. Ітерація. Інкрементальні та ітеративні моделі. Спіральна модель. Рекомендовані джерела: [1] ст.1-88
2.	Тема 1. 3. Етапи розробки програмного забезпечення (вимоги, проектування, реалізація, тестування). Збір та аналіз вимог. Проектування системи. Реалізація (програмування). Тестування Впровадження. Технічна підтримка та обслуговування Тема 1.4. Ролі та відповідальності на кожній фазі. Рекомендовані джерела: [1] ст.88-120, [2] ст.201-260
РОЗДІЛ 2 УСТАНОВКА ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
3	Тема 2.1 Процес установки програмного забезпечення. Підготовка системи (hardware/software), Установка програмного забезпечення, Конфігурація. Тестування після установки. Документація. Виправлення помилок. Оновлення та патчі. Підтримка користувачів Тема 2.2 Методи супроводу програмного забезпечення під час установки. Документація. Виправлення помилок. Тема 2.3 Супровід програмного забезпечення після установки Оновлення та патчі. Підтримка користувачів Рекомендовані джерела: [1] ст.88-120, [2] ст.201-260

РОЗДІЛ 3. МЕТОДИ ТА ТЕХНІКИ, ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ ПРОГРАМНОЇ ІНЖЕНЕРІЇ	
4	<i>Тема 3.1 Методи об'єктно-орієнтованої інженерії</i> <i>Метод об'єктного моделювання (OMT), Уніфікований процес розробки (RUP). Уніфікована мова моделювання (UML). Метод Джекобсона (OOSE). Метод Буча (Booch Method). Метод Шлаєра-Меллора (Shlaer-Mellor Method). Метод Кода і Йодона (Coad and Yourdon Method)</i> <i>Рекомендовані джерела: [3] ст.301-330</i>
5	<i>Тема 3.2 Методи функціонального програмування.</i> <i>Чисті функції (Pure Functions). Незмінність (Immutability). Вищі функції (Higher-Order Functions)</i> <i>Композиція функцій (Function Composition). Лінійні обчислення (Lazy Evaluation)</i> <i>Рекомендовані джерела: [3] ст.331-360</i>
6	<i>Тема 3.3 Техніки TDD, BDD, рефакторинг</i> <i>Test-Driven Development (TDD)- Написання тесту.Запуск тесту. Реалізація коду Запуск тесту зновуПеревірка, що тест проходить (зелена стадія).</i> <i>BDD -Написання сценаріїв. Реалізація тесту. Реалізація коду.</i> <i>Рефакторинг: Аналіз коду, внесення змін, перевірка, повторення</i> <i>Відмінності верифікації та валідації програмного забезпечення. Артефакти проекту створення програмного забезпечення. Верифікація проектного рішення, плану, вимог, тестових планів. Аудит програмного забезпечення. Експертиза. Методи статичного аналізу. Динамічні методи.</i> <i>Тема 3.4 Інструменти програмної інженерії та системи контролю версій</i> <i>Циклічні Інтегровані середовища розробки (IDE): Visual Studio, IntelliJ IDEA, Eclipse, PyCharm. Системи безперервної інтеграції та доставки (CI/CD). Інструменти для тестування. Інструменти для управління проектами. Інструменти для документування. Системи контролю версій</i>
7	<i>Тема 3.5 Мікросервісні технології програмної інженерії та хмарні обчислення</i> <i>Декомпозиція на сервіси. Комунікація через API. Незалежне розгортання. Технологічна різноманітність. Масштабованість. Стійкість до відмов Моделі хмарних обчислень включають: Інфраструктура як послуга (IaaS). Платформа як послуга (PaaS). Програмне забезпечення як послуга</i>
8	<i>Модульна контрольна робота 1</i>
РОЗДІЛ 4. УПРАВЛІННЯ ПРОГРАМНИМ ПРОЄКТОМ. ГРУПОВА РОЗРОБКА ПРОЄКТІВ ТА КЕЙСІВ	
9.	<i>Тема 4.1 Планування та оцінка проекту. Розподіл ресурсів, контроль якості та управління ризиками</i> <i>Визначення цілей і вимог проекту Розробка структури розбиття робіт (WBS). Розбиття проекту на менші, керовані завдання та підзавдання Оцінка тривалості та вартості завдань Розробка графіка проекту. Розробка плану управління ресурсами. Метод експертних оцінок. Метод аналогії. Метод параметричної оцінки. Метод трьох точок. Розподіл ресурсів. Ідентифікація ресурсів. Призначення ресурсів завданням. Оптимізація використання ресурсів. Моніторинг та контроль</i>

	використання ресурсів. Контроль якості. Управління ризиками. Моніторинг та контроль ризиків Рекомендовані джерела: 1-5, 10, 11 Завдання на СРС: Розробити план проекту з оцінкою ресурсів, ризиків та контролем якості.
10	Тема 4.2 Використання методологій Scrum та Kanban Scrum. Ролі в Scrum. Артефакти Scrum. Події Scrum Канбан. Візуалізація робочого процесу . Обмеження незавершених завдань (ліміти WIP). Управління потоком . Ясність та прозорість . Порівняння Scrum та Kanban Тема 4.3 Управління конфліктами Причини конфліктів. Різниця в цілях та перевагах. Обмежені ресурси. Різні стилі роботи. Комунікаційні проблеми. Особливості конфлікту. Типи конфліктів. Функціональні конфлікти. Дисфункціональні конфлікти. Стратегії управління конфліктами. Уникнення. Пристосування. Конкуренція (Competing). Компроміс (Компроміс). Співпраця. Процес управління конфліктами. Ідентифікація конфлікту. Аналіз конфлікту. Розробка стратегії вирішення конфлікту. Впровадження стратегії. Моніторинг та оцінка результатів Рекомендовані джерела: 1-5, 10, 11 Завдання на СРС Спроекувати Kanban-дошку для управління проектом розробки ПЗ.
РОЗДІЛ 5. ЕТИЧНІ АСПЕКТИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ	
11	Тема 5.1. Відповідальність розробників ПЗ, законодавча та нормативна Законодавча відповідальність. Дотримання авторських прав і ліцензійних угод Виконання вимог щодо конфіденційності та захисту даних Дотримання стандартів безпеки інформації Відповідність законодавству про захист споживачів Виконання вимог щодо доступності програмного забезпечення Нормативна відповідальність Дотримання галузевих стандартів і норм Використання етичних принципів у розробці ПЗ Забезпечення якості та надійності програмного забезпечення Відповідальність за наслідки використання ПЗ Підтримка та оновлення програмного забезпечення Рекомендовані джерела: 1, 3, 5, 7, 9, 13
12	Тема 5.2. Приватність та безпека даних Приватність даних. Захист персональних даних користувачів. Анонімізація та псевдонімізація даних. Дотримання законодавства про захист даних (наприклад, GDPR) Політика конфіденційності та інформування користувачів. Обмеження доступу до особистої інформації. Безпека даних. Шифрування даних під час передачі та зберігання Аутентифікація та авторизація користувачів. Виявлення та запобігання витокам даних Захист від шкідливого програмного забезпечення та кібератак
13	Тема 5.3. Відповідальне використання штучного інтелекту

	<i>Завдання на СРС Проаналізувати етичні аспекти використання AI-технологій у ПЗ.</i>
РОЗДІЛ 6. НОВІТНІ ТЕНДЕНЦІЇ В ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
14	<i>Тема 6.1 Штучний інтелект та машинне навчання. Управління даними та конфіденційність. Регулярне тестування та валідація моделей. Виявлення та усунення упереджень в алгоритмах. Моніторинг та аудит систем ШІ. Законодавчі та нормативні аспекти. Дотримання існуючих законів і регуляцій. Розробка нових нормативних актів для регулювання ШІ. Співпраця з регуляторними органами. Відповідальність за порушення законодавства. Інформування та освіта користувачів щодо використання ШІ</i>
15	<i>Тема 6.2 Поняття DevOps, CI/CD. DevOps: Інтеграція розробки та операцій. Автоматизація процесів розгортання. Культура співпраці між командами. Постійний моніторинг та зворотний зв'язок. Підвищення швидкості та якості випуску програмного забезпечення. CI: Постійне інтегрування змін в основну гілку коду. Автоматичне тестування всіх змін. Використання системи контролю версій. Швидке виявлення та виправлення помилок. Зниження ризиків при інтеграції нових функцій. CD: Автоматизація процесу доставки програмного забезпечення. Постійна готовність до випуску нової версії. Автоматичне розгортання на іншому середовищі (тестове, продуктивне). Забезпечення стабільності та якості випуску. Рекомендовані джерела: 1, 3, 5, 7, 9, 13</i>
16	<i>Тема 6.3 Технології віртуальної та доповненої реальності. Віртуальна реальність (VR). Повне занурення користувача у віртуальне середовище. Використання VR-гарнітури (наприклад, Oculus Rift, HTC Vive) Сфери застосування: Відеоігри. Освіта та тренінги. Медицина. Архітектура та дизайн. Технічні аспекти: Висока роздільна здатність дисплеїв. Трекінг рухів голови та тіла. Реалістична графіка та звук. Виклики та обмеження: Висока вартість обладнання. Обмежений час використання через можливості проблеми зі здоров'ям (наприклад, віртуальна хвороба). Доповнена реальність (AR). Накладання цифрової інформації на реальний світ. Використання AR-гарнітури (наприклад, Microsoft HoloLens) та мобільних пристроїв. Сфери застосування: Маркетинг та реклама. Навігація та туризм. Освіта та тренінги. Промисловість та ремонт. Технічні аспекти: Висока точність трекінгу. Інтеграція з камерами та сенсорами. Використання комп'ютерного зору. Виклики та обмеження: Вимоги до обчислювальних ресурсів. Проблеми з конфіденційністю та безпекою даних. Складність розробки та інтеграції AR-додатків</i>
17	<i>Модульна контрольна робота 2</i>
18	<i>Залік.</i>

Лабораторні заняття

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	Вступ до предмету	2

2	<i>Лабораторна робота 1 Аналіз вимог для енергетичної системи. Практичне застосування теоретичного матеріалу за темою лабораторної роботи</i> <i>Мета: Вивчити процес збору та аналіз вимог до програмного забезпечення для управління енергетичними системами.</i> <i>Завдання: Провести інтерв'ю з фахівцями в галузі енергетики та скласти документ з вимогами до ПЗ для моніторингу та управління електромережами</i>	2
3	<i>Лабораторна робота 1 Аналіз вимог для енергетичної системи.</i> <i>Мета: Вивчити процес збору та аналіз вимог до програмного забезпечення для управління енергетичними системами.</i> <i>Завдання: Провести інтерв'ю з фахівцями в галузі енергетики та скласти документ з вимогами до ПЗ для моніторингу та управління електромережами</i>	2
4	<i>Лабораторна робота 2. Установка програмного забезпечення для енергетичних систем. Практичне застосування теоретичного матеріалу за темою лабораторної роботи</i> <i>Мета: Вивчити процес установки програмного забезпечення управління енергетичними ресурсами.</i> <i>Завдання: Створити інсталятор для ПЗ, яке передбачає моніторинг та управління енергетичними ресурсами, та задокументувати процес встановлення.</i>	2
5	<i>Лабораторна робота 2. Установка програмного забезпечення для енергетичних систем.</i> <i>Мета: Вивчити процес установки програмного забезпечення управління енергетичними ресурсами.</i> <i>Завдання: Створити інсталятор для ПЗ, яке передбачає моніторинг та управління енергетичними ресурсами, та задокументувати процес встановлення.</i>	2
6	<i>Лабораторна робота 3: Розробка мікросервісів для енергетичних систем</i> <i>Практичне застосування теоретичного матеріалу за темою лабораторної роботи</i> <i>Мета: Вивчити принципи розробки мікросервісів для енергетичних систем.</i> <i>Завдання: Реалізувати просту систему на основі мікросервісної архітектури для моніторингу та управління енергетичними ресурсами.</i> <i>Лабораторна робота 5: Планування проекту в енергетиці</i>	2
7	<i>Лабораторна робота 3: Розробка мікросервісів для енергетичних систем</i> <i>Мета: Вивчити принципи розробки мікросервісів для енергетичних систем.</i> <i>Завдання: Реалізувати просту систему на основі мікросервісної архітектури для моніторингу та управління енергетичними ресурсами.</i> <i>Лабораторна робота 5: Планування проекту в енергетиці</i>	2
8	<i>Лабораторна робота 4. Використання машинного навчання в енергетиці</i> <i>Практичне застосування теоретичного матеріалу за темою лабораторної роботи</i> <i>Мета: Ознайомитися з використанням машинного навчання для оптимізації енергетичних систем.</i> <i>Завдання: Реалізувати алгоритм машинного навчання для прогнозування споживання енергії на основі історичних даних.</i>	2
9	<i>Лабораторна робота 4. Використання машинного навчання в енергетиці</i> <i>Мета: Ознайомитися з використанням машинного навчання для оптимізації енергетичних систем.</i>	2

	<i>Завдання: Реалізувати алгоритм машинного навчання для прогнозування споживання енергії на основі історичних даних.</i>	
--	---	--

Самостійна робота

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	<i>Розділ 1, Тема 1.1 Порівняльний аналіз моделей життєвого циклу програмного забезпечення (каскадна, ітераційна, спіральна тощо): Переваги та недоліки кожної моделі. Ситуації, в яких доцільно застосовувати ту чи іншу модель. Приклади реальних проектів, де використовувалися різні моделі</i> Завдання на СРС: Порівняти та проаналізувати різні моделі життєвого циклу ПЗ (каскадна, ітераційна, спіральна тощо)	7
2	<i>Розділ 1, Тема 1.3 Розробка плану проекту з чітким визначенням вимог, проектування, реалізації та тестування:Визначення цілей та вимог проекту. Структура розбиття робіт (WBS). Оцінка тривалості та вартості завдань. Розробка графіка проекту. Розподіл ресурсів</i> Завдання на СРС: Розробити план проекту з чітким визначенням вимог, проектування, реалізації та тестування	7
3	<i>Розділ 2, Тема 2,1. Створення покрокової інструкції з установки програмного забезпечення, включаючи перевірку працездатності: Підготовка системи (апаратне та програмне забезпечення).Процес установки ПЗ.Конфігурація системи.Тестування після установки Документація та підтримка користувачів</i> Завдання на СРС: Розробити покрокову інструкцію з установки ПЗ, включаючи перевірку працездатності.	8
4	<i>Застосування технік TDD та BDD при розробці невеликого програмного модуля:Написання тестів. Реалізація коду. Рефакторинг</i> Порівняння TDD та BDD Завдання: Застосувати техніки TDD та BDD при розробці програмного модуля.	7
6	<i>Розділ 3, Тема 3.4. Розробка плану проекту з оцінкою ресурсів, ризиків та контролем якості:Ідентифікація та оцінка ресурсів.Управління ризиками</i> Контроль якості Завдання на СРС: Налаштувати систему контролю версій (наприклад, Git) для командної розробки ПЗ	7
7	<i>Розділ 4, Тема 4.2 Проектування Kanban-дошки для управління проектом розробки ПЗ: Візуалізація робочого процесу. Обмеження незавершених завдань. Управління потоком. Порівняння Scrum та Kanban</i> Завдання на СРС Спроектувати Kanban-дошку для управління проектом розробки ПЗ.	8
8	<i>Розділ 5, Тема 5.2. Розробка політики безпеки та конфіденційності для ПЗ, що обробляє персональні дані:Захист персональних даних користувачів</i> Аутентифікація та авторизація. Безпека даних (шифрування, запобігання витокам). Відповідність законодавству (наприклад, GDPR) Завдання на СРС: Розробити політику безпеки та конфіденційності для ПЗ, що обробляє персональні дані.	8
9	<i>Розділ 6, Тема 6.1. Аналіз етичних аспектів використання AI-технологій у програмному забезпеченні: Управління даними та конфіденційність. Виявлення та усунення упереджень. Законодавчі та нормативні аспекти. Відповідальність розробників</i> Завдання на СРС: [1] с. 22, [2] с. 117-122.	8
10	<i>Залік</i>	6

6. Політика навчальної дисципліни (освітнього компонента)

Для успішного проходження курсу та складання контрольних заходів необхідним є вивчення навчального матеріалу за кожною темою. Специфіка курсу передбачає акцент на розумінні підходів і принципів, отримання практичних навичок, а не просто запам'ятовування визначень. Кожен студент повинен ознайомитися і слідувати Положенню про академічну доброчесність, Статуту і розпорядку дня університету та дотримуватись Політики використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського. Для успішного засвоєння програмного матеріалу студент зобов'язаний:

- не запізнюватися на заняття;
- не пропускати заняття, а в разі пропуску відновити за допомогою консультування з викладачем та з використанням конспекту на платформі дистанційного навчання «Сікорський», самостійно вивчити матеріал пропущеного заняття та скласти відповідні контрольні заходи розміщені на платформі дистанційного навчання «Сікорський в індивідуальному порядку;
- конструктивно підтримувати зворотній зв'язок на всіх заняттях;
- брати активну участь у освітньому процесі;
- своєчасно і старанно виконувати завдання для самостійної роботи;
- бути доброзичливим до однокурсників та викладачів;
- брати участь у контрольних заходах;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватися індивідуально (в дистанційній online формі за погодженням із деканом факультету);
- будь-яке копіювання або відтворення результатів чужої праці (у тому числі списування), якщо тільки робота не має груповий формат, використання чужих завантажених з Інтернету матеріалів кваліфікується як порушення норм і правил академічної доброчесності та передбачає притягнення винного до відповідальності, у порядку, визначеному чинним законодавством та Положенням про академічну доброчесність університету. Результатом невиконання та/або недотримання правил може бути оцінка «не зараховано» за курс.

Система вимог, які викладач ставить перед студентом:

- Кодекс честі: <http://kpi.ua/code>;
- Правила внутрішнього розпорядку: <http://kpi.ua/admin-rule>;
- Положення про організацію освітнього процесу в КПІ ім. Ігоря Сікорського: <https://kpi.ua/regulations>
- Політика використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського https://osvita.kpi.ua/sites/default/files/downloads/polityka-vykorystannia-shtuchnogo-intelektu_2023.pdf

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: тестування на платформі дистанційного навчання «Сікорський», МКР, виконання самостійних завдань та лабораторних робіт.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: семестровий рейтинг більше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
95-100	Відмінно
85-94	Дуже добре
75-84	Добре
65-74	Задовільно
60-64	Достатньо
Менше 60	Незадовільно

Менше 30	Не допущено
----------	-------------

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- тестування на платформі дистанційного навчання «Сікорський» по лекційним заняттям;
- виконання лабораторних робіт;
- виконання модульної контрольної роботи (МКР);
- виконання і захист самостійних домашніх завдань;
- виконання додаткових завдань.

Тестування по лекціям	Лабораторні роботи	МКР	Захист самостійного домашнього завдання	Додаткові бали
18	40	16	26	10

Тестування по матеріалам лекційних занять

Ваговий бал - 1. Максимальна кількість балів за тестування – 1 бал * 18 лекцій = 18 балів.

Тестування може проводитися за допомогою систем дистанційного навчання, наприклад Moodle, яке доступне протягом 2 робочих днів після завершення поточної лекції. У деяких випадках термін проходження тестування може бути продовжений лектором. Тривалість проходження одного тестування – 10 хвилин. Кількість спроб – одна. У деяких випадках, що пов'язані з технічними проблемами студентів, може надатися повторна спроба на окремі тестування.

Кожне тестування містить 10 запитань різного формату (вибір правильного варіанту з переліку; вірно/невірно; визначити відповідність; чисельна відповідь; вибір пропущених слів; перетаскування на зображення тощо).

Критерії оцінювання

- запитання типу «вибір правильного варіанту з переліку», «вірно/невірно», «чисельна відповідь» оцінюються однозначно: вірна відповідь – 0,1 бал, невірна відповідь – 0 балів;
- запитання, на які немає однієї конкретної відповіді, типу «визначити відповідність», «вибір пропущених слів», «перетаскування на зображення» оцінюються у відповідності до кількості елементів у тесті (наприклад, якщо треба вставити 4 слова у текст, то студент отримає по 0,025 балів за одне правильне вставлене слово, а за всі 4 правильно вставлені слова отримає відповідно 0,1 бал) – невірна відповідь – 0 балів, частково вірна відповідь – 0,01-0,09 балів, вірна відповідь 0,1 бал.

Практичні роботи

Ваговий бал –10. Максимальна кількість балів за всі практичні заняття – 10 бали * 4 заняття= 40 балів.

На лабораторних роботах студенти перевіряють працездатність написаних програм або схем за попередньо вирішеними вдома задачами. Для допуску до поточної лабораторної роботи необхідно мати Протокол, оформлений відповідно до норм оформлення технічної документації, який має містити всі необхідні пункти, відповідно до Методичних вказівок. Також для допуску до лабораторної роботи (окрім 1-ї) необхідно захистити попередню. Студенти, що не захистили попередню лабораторну роботу можуть бути не допущені до виконання наступної. Лабораторні роботи виконуються кожним студентом індивідуально.

Перше лабораторне заняття, зазвичай, поводить, коли лекційний матеріал ще не начитаний, тому його тематика не пов'язана з конкретними темами дисципліни, а направлена на перевірку логічного мислення студентів та можливості інтуїтивно, без знань методів синтезу, скласти схеми для простих логічних задач.

Критерії оцінювання лабораторної роботи

Критерії оцінювання лабораторної роботи з ваговим балом 10:

- *вірно виконаний синтез всіх задач, продемонстрована працездатність всіх програм (схем), вірні відповіді на запитання до захисту – 9-10 балів;*
- *вірно виконаний синтез всіх задач, продемонстрована працездатність всіх програм (схем), відповіді на питання до захисту мають неточності – 6-8 балів;*
- *виконаний синтез всіх задач, але деякі з них містять помилки або неточності, продемонстрована працездатність не всіх програм (схем), відповіді на питання до захисту мають неточності – 3-5 балів;*
- *виконаний синтез не всіх задач, продемонстрована працездатність не всіх програм (схем), відповіді на питання до захисту мають неточності – 0-2 балів;*
- *лабораторна робота не виконана або протокол не представлений – повертається на відпрацювання або доопрацювання.*

Модульна контрольна робота

Ваговий бал МКР – 16. Максимальний бал за МКР складає 16 балів.

Модульна контрольна робота складається з 2-х частин. Кожна оцінюється в 8 балів.

На модульній контрольній роботі студент має виконати 2 завдання. Завдання оцінюються від 0 до 4 балів в залежності від правильності вирішення.

Критерії оцінювання

- *завдання вирішено вірно та здано протягом 2-х тижнів після практичного заняття – 4 балів;*
- *завдання вирішено вірно, але здано протягом більш ніж 2-х тижнів після практичного заняття – 3 бал;*
- *завдання вирішено із незначними помилками та здано протягом 2-х тижнів після практичного заняття – 2 балів;*
- *завдання вирішено із незначними помилками та здано протягом більш ніж 2-х тижнів після практичного заняття – 1 балів;*
- *домашнє завдання вирішено із значними помилками – повертається на доопрацювання.*

Захист індивідуального домашнього завдання.

Ваговий бал – 26. Максимальний бал за захист індивідуального домашнього завдання складає 26 балів.

Студент представляє презентацію проекту по 9 домашнім завданням. Оцінюється проект в загальному.

Критерії оцінювання

1. *Рейтинг захисту $R_z = 24 - 26$ балів – повністю виконані всі завдання, своєчасно оформлено та представлено супровідну пояснювальну записку. При захисті грамотно та логічно послідовно викладено основні положення роботи у вигляді доповіді, в процесі відповідей на питання продемонстрував наявність глибоких вичерпних знань, або твердих та достатньо повних знань.*
2. *Рейтинг захисту $R_z = 18 - 23$ балів – відповідаючи на питання під час презентації власної роботи, здобувач припускається окремих помилок, але може їх виправити за допомогою викладача, знає визначення основних понять і величин, впевнено орієнтується в своїй роботі.*
3. *Рейтинг захисту $R_z = 14-17$ балів – здобувач відповідає майже на всі питання під час презентації власної роботи. Відповіді іноді непослідовні та нечіткі. Своєчасно оформив та представив презентацію власного проекту.*
4. *Рейтинг захисту $R_z = 10-13$ балів – здобувач частково відповідає на всі питання під час презентації власної роботи, показує знання, але не впевнено орієнтується в своїй роботі. Відповіді непослідовні та нечіткі. Не своєчасно оформив та представив презентацію власного проекту.*
5. *Рейтинг захисту $R_z = 5 - 9$ балів – здобувач частково відповідає на деякі питання під час презентації власної роботи, показує незадовільні знання. Відповіді непослідовні та нечіткі. Не своєчасно оформив та представив презентацію власного проекту.*

6. Рейтинг захисту $R_3 < 5$ балів – у відповіді здобувач припускається суттєвих помилок, не може виправити помилки за допомогою викладача. Відповіді некоректні, а в деяких випадках не відповідають суті поставленого питання. Не своєчасно оформив та представив презентацію власного проєкту.

Додаткові (бонусні) бали

Рейтинговою системою оцінювання передбачені додаткові бали за виконання додаткових завдань. Один студент не може отримати більше ніж 10 бонусних балів у семестрі. При отриманні більш ніж 10 балів, вони обмежуються на рівні 10. Бонусні бали можуть бути отримані за такі види робіт: «Івенти», «Додаткові лекції» та «Завдання до лекцій».

Івенти. Івенти - це спеціальні події для студентів, які хочуть отримати додаткові бали за вирішення ускладнених завдань. Івенти активуються у визначений час і активні обмежений час. Додаткові бали отримують тільки ті студенти, які надали правильну відповідь та вчасно її завантажили. Кількість балів за додаткові завдання визначає кожен івент окремо. Один студент не може отримати більш ніж 10 балів за івенти.

Додаткові лекції. Самостійна робота студентів передбачає до 10 додаткових лекцій, які студенти повинні опрацювати та законспектувати. За опрацювання однієї лекції вигляді у конспекту нараховується 1 бал. Максимальна кількість балів, що можна отримати за опрацювання додаткових лекцій складає 5 балів.

Завдання до лекцій. Студенти, за бажанням, можуть виконувати додаткові завдання за матеріалами лекцій (розв'язати приклад, зробити доповідь тощо). За одне додаткове завдання нараховується 0,5 бали. Максимальна кількість балів, що можна отримати за завдання до лекцій складає 5 балів.

Форма семестрового контролю – залік

Максимальна сума балів складає 100. Необхідною умовою допуску до заліку є зарахування всіх домашніх робіт та робота на практичних заняттях, а також стартовий рейтинг. Для отримання заліку з кредитного модулю «автоматом» потрібно мати рейтинг не менше 60 балів, а також виконані умови допуску до заліку.

Здобувачі, які наприкінці семестру мають рейтинг менше 60 балів, а також ті, хто хоче підвищити свою оцінку в системі ECTS, виконують залікову контрольну роботу. При цьому набрані бали студентом анулюються, а оцінка за залікову контрольну роботу є остаточною.

Залікова робота. Залікова робота проводиться на останньому лекційному занятті. Здобувач проходить тестування очного або у середовищі дистанційного навчання, наприклад Moodle. На тестування пропонується 100 тестових, кожне з яких оцінюється в 1 бал. Для отримання позитивної оцінки необхідно набрати 60 балів і вище. Час тестування зазвичай складає 100 хвилин, але може бути скоригований лектором та (або) викладачам, що приймає залік.

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий:

1. Які основні моделі життєвого циклу програмного забезпечення ви знаєте? Порівняйте їх переваги та недоліки.
2. Поясніть, в яких ситуаціях доцільно застосовувати каскадну, ітераційну та спіральну моделі розробки ПЗ.
3. Опишіть основні етапи розробки програмного забезпечення (вимоги, проєктування, реалізація, тестування).
4. Які ролі та відповідальності існують на кожній фазі розробки ПЗ?
5. Розкрийте процес установки програмного забезпечення, включаючи підготовку системи, конфігурацію, тестування та документацію.
6. Охарактеризуйте методи супроводу програмного забезпечення під час та після установки.

7. Які методи об'єктно-орієнтованої інженерії ви знаєте? Наведіть приклади та порівняйте їх.
8. Поясніть основні концепції функціонального програмування: чисті функції, незмінність, вищі функції, композиція функцій, лінійні обчислення.
9. Розкрийте суть технік Test-Driven Development (TDD) та Behavior-Driven Development (BDD). Наведіть приклади їх застосування.
10. Охарактеризуйте відмінності між верифікацією та валідацією програмного забезпечення.
11. Які інструменти програмної інженерії та системи контролю версій ви знаєте? Як налаштувати систему контролю версій для командної розробки ПЗ?
12. Поясніть основні принципи мікросервісної архітектури та переваги хмарних обчислень.
13. Як здійснюється планування та оцінка проекту розробки ПЗ? Опишіть методи оцінки тривалості та вартості завдань.
14. Розкрийте процес розподілу ресурсів, контролю якості та управління ризиками в проекті.
15. Порівняйте методології Scrum та Kanban. Як вони допомагають в управлінні проектами розробки ПЗ?
16. Охарактеризуйте основні причини, типи та стратегії управління конфліктами в команді розробників.
17. Які законодавчі та нормативні аспекти відповідальності розробників ПЗ ви знаєте?
18. Поясніть, як забезпечити приватність та безпеку даних у програмному забезпеченні.
19. Розкрийте етичні аспекти використання штучного інтелекту в програмних системах.
20. Що таке DevOps? Поясніть принципи безперервної інтеграції (CI) та безперервної доставки (CD).
21. Охарактеризуйте технології віртуальної та доповненої реальності, їх сфери застосування та технічні аспекти.
22. Які основні виклики та обмеження існують у використанні VR та AR технологій?
23. Поясніть, як здійснюється управління даними та забезпечення конфіденційності в системах на основі штучного інтелекту.
24. Як можна виявляти та усувати упередженість в алгоритмах ШІ?
25. Які законодавчі та нормативні аспекти необхідно враховувати при розробці систем на основі ШІ?
26. Як забезпечити відповідальне використання ШІ-технологій у програмному забезпеченні?
27. Охарактеризуйте основні артефакти проекту створення програмного забезпечення.
28. Поясніть, як здійснюється верифікація проектного рішення, плану, вимог та тестових планів.
29. Які методи статичного та динамічного аналізу програмного забезпечення ви знаєте?
30. Як здійснюється аудит та експертиза програмного забезпечення?
31. Опишіть основні принципи та практики безперервної інтеграції та безперервної доставки.
32. Які інструменти для управління проектами, тестування та документування ви знаєте?
33. Поясніть, як використовуються системи контролю версій у процесі розробки ПЗ.
34. Охарактеризуйте основні моделі хмарних обчислень: IaaS, PaaS, SaaS.
35. Які переваги та недоліки мікросервісної архітектури порівняно з монолітною?
36. Розкрийте процес управління ризиками в проекті розробки програмного забезпечення.
37. Поясніть, як здійснюється контроль якості на різних етапах розробки ПЗ.
38. Опишіть основні етичні принципи, які необхідно враховувати при розробці програмного забезпечення.

Робочу програму навчальної дисципліни (Силабус) дисципліна «Методологія інженерії програмного забезпечення / Software Engineering Methodology»:

Складено доцентом кафедри інженерії програмного забезпечення в енергетиці НН ІАТЕ, к.т.н. Залевською Ольгою Валеріївною

Ухвалено кафедрою ІПЗЕ (протокол № 42 від 18.06.2025)

Погоджено Методичною комісією НН ІАТЕ (протокол № 9 від 27.06.2025)