

Код та назва дисципліни українською мовою/ Назва дисципліни англійською мовою	1-ф09-4_ Основи об'єктно-орієнтованого аналізу і розробки інформаційних систем Fundamentals of Object-Oriented Analysis and Information Systems Development
Рекомендується для галузі знань (спеціальності, освітньої програми)	Для спеціальностей усіх галузей знань
Кафедра (зазначати повну назву кафедри)	Електронних обчислювальних машин
П.І.П. НПП (за можливості)	Карпенко Надія Валеріївна або Литвинов Олександр Анатолійович
Рівень ВО	Бакалавр
Курс, семестр (в якому буде викладатись)	3-4 курс. Для 4 курсу тільки 1 семестр.
Мова викладання	Українська
Пререквізити (передумови вивчення дисципліни)	Вміння спілкуватися, допитливість, володіння офісними пакетами програм
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> ✓ Дисципліна є актуальною, оскільки об'єктно-орієнтований підхід є основою для більшості сучасних мов програмування (Java, C++, C#, Python тощо) та архітектур програмних систем. ✓ Дисципліна вчить аналізувати вимоги, проектувати архітектуру системи, створювати UML-діаграми — це ключові навички аналітика та архітектора ПЗ. ✓ Документовані вимоги – це те, без чого не починається розробка програмного забезпечення, вони потрібні для розробників, тестувальників, аналітиків і замовника. Це мінімізує непорозуміння.
Перелік тем з дисциплін	<ul style="list-style-type: none"> ✓ Поняття product vision. Визначення цільової аудиторії та основної мети продукту. Основні функції та переваги продукту. ✓ Види вимог. Методи збору вимог. Стейкхолдери. ✓ User Stories, Use Cases. ✓ Виявлення суперечностей і неоднозначностей. Пріоритезація вимог. Трасування вимог. ✓ Роль бізнес-аналітика і project-менеджера у проекті. ✓ Ідентифікація сутностей та їхніх взаємозв'язків. Побудова об'єктної моделі предметної області. Виділення бізнес-об'єктів і процесів. ✓ Огляд уніфікованої мови моделювання (UML). Типи UML-діаграм. ✓ Моделювання структури системи. ✓ Моделювання поведінки. ✓ Розбиття системи на підсистеми. Архітектурні патерни. ✓ Роль тестувальника в процесі аналізу вимог. Методика перетворення вимог у тест-кейси. ✓ Приклади тест-кейсів для функціональних вимог. ✓ Трасування вимог до тестових сценаріїв.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Знання і уміння:</p> <ul style="list-style-type: none"> ✓ збирати, аналізувати та чітко формулювати функціональні та нефункціональні вимоги, використовуючи сучасні підходи; ✓ створювати структуру і логіку програмного

	<p>забезпечення через діаграми UML;</p> <ul style="list-style-type: none"> ✓ створювати тест-кейси на основі вимог і перевіряти відповідність реалізації початковим очікуванням
Очікувані результати навчання	<p>Окрім теоретичних знань студент набуває практичні навички:</p> <ul style="list-style-type: none"> ✓ збору, класифікації й аналізу вимог до ПЗ; ✓ створення специфікацій програмного забезпечення; ✓ побудови моделей структури та поведінки інформаційної системи за допомогою діаграм UML; ✓ роботи з інструментами моделювання (Visual Paradigm, Draw.io) та інструментами керування проектами (Jira);
Інформаційне забезпечення	Конспект лекцій, методичні вказівки
Види навчальних занять (<i>лекції, практичні, семінарські, лабораторні заняття тощо</i>)	Лекції (28 год.), лабораторні заняття (28 год.)
Вид семестрового контролю	диф. залік
Максимальна кількість здобувачів на семестр/ Мінімальна кількість здобувачів (<i>тільки для мовних, творчих дисциплін, за необхідності</i>)	Без обмежень

Декан факультету _____

Ігор ГОМІЛКО