МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету
_____ Інна КОНДЮС
«29» серпня 2026 р.

ЗАТВЕРДЖУЮ
Перший проректор
_____ Надія КОВАЛЬЧУК
«29» серпня 2026 р.

СИЛАБУС

з освітнього компонента «Об'єктно-орієнтоване програмування»
ступінь вищої освіти – бакалавр
галузь знань – F Інформаційні технології
спеціальність – F2 Інженерія програмного забезпечення
освітня програма – Інженерія програмного забезпечення

Кафедра інженерії програмного забезпечення

Силабус склав(ла) доцент кафедри інженерії програмного забезпечення, к.т.н., доцент Андрій ЯЩУК

Силабус затверджений на засіданні кафедри інженерії програмного забезпечення, протокол № 1/1 від «28» серпня 2026 р.

Завідувач кафедри к.т.н., доцент Наталія ЛІЩИНА

Розглянуто та схвалено групою забезпечення освітньої програми «Інженерія програмного забезпечення», протокол № 2 від «28» серпня 2026 р.

Гарант ОП «Інженерія програмного забезпечення» к.т.н., доцент Наталія ЛІЩИНА

Схвалено науково-методичною радою університету, протокол № 1/а від «29» серпня 2026 р.

Затверджений силабус онлайн:
https://e.lntu.edu.ua/syllabus/1305

Луцьк – 2026

СИЛАБУС ОСВІТНЬОГО КОМПОНЕНТА «Об'єктно-орієнтоване програмування»

Освітня програма	Інженерія програмного забезпечення
Галузь знань / Спеціальність	F Інформаційні технології / F2 Інженерія програмного забезпечення
Рівень вищої освіти	перший (бакалаврський)
Статус дисципліни	нормативна
Мова викладання	українська
Обсяг	7 кредитів ECTS / 210 годин, семестр 3
Форма контролю	екзамен, курсова робота
Викладач(і)	к.т.н., доцент Андрій ЯЩУК електронна пошта: a.yashchuk@lutsk-ntu.com.ua

1. Мета і завдання дисципліни

Мета: Формування у здобувачів цілісного розуміння принципів об’єктно-орієнтованого програмування та практичних умінь проєктувати, реалізовувати, налагоджувати й супроводжувати застосунки мовою C# на платформі .NET із використанням сучасних засобів роботи з даними, графічним інтерфейсом, мережею, файлами та базами даних.

Завдання: Засвоїти поняття, принципи та сфери застосування об’єктно-орієнтованої парадигми.
Опанувати синтаксис C# і базові компоненти платформи .NET.
Навчитися створювати класи, об’єкти та ієрархії типів, застосовувати інкапсуляцію, успадкування, поліморфізм, абстрактні класи й інтерфейси.
Сформувати навички використання узагальнень, колекцій, делегатів, лямбда-виразів, подій і LINQ.
Навчитися створювати подієво-керовані графічні застосунки, виконувати асинхронні мережеві операції, опрацьовувати JSON, файли й винятки.
Опанувати основи збереження даних у SQLite засобами Entity Framework Core та виконання CRUD-операцій.
Розвинути навички декомпозиції задач, налагодження, тестування, усного пояснення програмних рішень і самостійного пошуку технічної інформації.

Пререквізити: Алгоритми та структури даних, Основи програмування, Програмування

2. Компетентності

ІК — інтегральна компетентність: 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.

ЗК01 — Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02 — Здатність застосовувати знання у практичних ситуаціях.

ЗК05 — Здатність вчитися і оволодівати сучасними знаннями.

СК01 — Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК03 — Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК11 — Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК13 — Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

3. Структура дисципліни, розподіл годин, зв'язок тем з компетентностями

№	Назва теми	**Лекції** денна/ заочна	**Лаб.** денна/ заочна	**Самост. робота** денна/ заочна	**Разом**	**Методи навчання**	**Методи оцінювання**	**ЗК та СК**
Змістовий модуль 1. Основи C#, .NET та об’єктно-орієнтованого програмування								
1	Тема 1. Парадигми програмування. Вступ до ООП	2/0,5	3/0,75	7/11,25	12/12,5	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК05, ЗК02, СК13, СК01, СК03, СК11
2	Тема 2. Мова C# і платформа .NET	2/0,5	3/0,75	7/11,25	12/12,5	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК05, ЗК01, ЗК02, СК13, СК01, СК03, СК11
3	Тема 3. Основи мови C#	2/0,5	3/0,5	7/11,25	12/12,25	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК01, ЗК05, СК13
4	Тема 4. Класи, об’єкти, структури та простори імен	2/0,5	3/0,5	7/11,25	12/12,25	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
5	Тема 5. Методи та декомпозиція програми	2/0,5	3/0,5	7/11	12/12	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
6	Тема 6. Успадкування	2/0,5	3/0,5	7/11	12/12	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03

№	Назва теми	**Лекції** денна/ заочна	**Лаб.** денна/ заочна	**Самост. робота** денна/ заочна	**Разом**	**Методи навчання**	**Методи оцінювання**	**ЗК та СК**
7	Тема 7. Віртуальні методи, абстрактні класи та поліморфізм	2/0,5	3/0,5	7/11	12/12	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
8	Тема 8. Інтерфейси	2/0,5	3/0,5	7/11	12/12	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
Змістовий модуль 2. Сучасні засоби C# і розроблення прикладних програм								
9	Тема 9. Типи даних, узагальнення та колекції	2/0,5	3/0,5	7/11	12/12	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
10	Тема 10. Делегати, лямбда-вирази та події	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК05, ЗК01, СК03, СК13
11	Тема 11. LINQ і декларативна обробка даних	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК01, ЗК02, ЗК05, СК03
12	Тема 12. Графічний інтерфейс користувача в .NET	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК05, ЗК01, СК03, СК13
13	Тема 13. Мережеві запити, JSON та асинхронне програмування	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК05, ЗК01, СК03, СК13

№	Назва теми	**Лекції** денна/ заочна	**Лаб.** денна/ заочна	**Самост. робота** денна/ заочна	**Разом**	**Методи навчання**	**Методи оцінювання**	**ЗК та СК**
14	Тема 14. Обробка винятків, файли та потоки даних	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК05, ЗК01, СК03, СК11, СК13
15	Тема 15. SQLite, Entity Framework Core та CRUD	2/0,25	3/0,5	7/11	12/11,75	МН1, МН2, МН3, МН6	МО1, МО2, МО4, МО7, захист лабораторної роботи	ЗК02, ЗК05, ЗК01, СК03, СК11, СК13
	Курсова робота			30/30	30/30			
	Разом	**30/6**	**45/8**	**135/196**	**210/210**			

Легенда: ЗК — загальна компетентність (номер за п. 2); СК — спеціальна (фахова) компетентність (номер за п. 2). Повне формулювання кожного коду наводиться в ОП / стандарті спеціальності.

Методи навчання: МН1 — словесний метод (лекція, дискусія, співбесіда тощо); МН2 — практичний метод (лабораторні та практичні заняття); МН3 — наочний метод (метод ілюстрацій і метод демонстрацій); МН6 — самостійна робота (розв'язання завдань).

Методи оцінювання: МО1 — екзамен; МО2 — усне або письмове опитування; МО4 — тестування; МО7 — презентації результатів виконаних завдань та досліджень.

4. Результати навчання

ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН08. Вміти розробляти людино-машинний інтерфейс.

ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

Після успішного завершення дисципліни здобувачі зможуть:
• пояснювати основні парадигми програмування та принципи ООП;
• створювати, запускати й налагоджувати проєкти C#/.NET;
• проєктувати класи та їх взаємодію, застосовувати інкапсуляцію, успадкування, поліморфізм, абстрактні класи й інтерфейси;
• використовувати узагальнення, колекції, делегати, події та LINQ для типобезпечної обробки даних;
• створювати графічні застосунки, отримувати дані з веб-API та виконувати тривалі операції асинхронно без блокування інтерфейсу;
• обробляти винятки, читати й зберігати дані у файлах JSON і CSV;
• працювати з SQLite через Entity Framework Core і реалізовувати CRUD-операції;
• обґрунтовувати прийняті проєктні рішення та самостійно пояснювати роботу створеного коду.

5. Цілі сталого розвитку

ЦСР 9 – Промисловість, інновації та інфраструктура

6. Неформальна освіта

Рекомендовано опрацьовувати навчальні модулі Microsoft Learn із C#, .NET та Entity Framework Core, тематичні курси, вебінари й тренінги з розроблення програмного забезпечення. Результати, здобуті у формальній, неформальній чи інформальній освіті, можуть бути визнані відповідно до процедур ЛНТУ. Підтвердженнями можуть бути сертифікат, опис досягнутих результатів і, за потреби, коротке верифікаційне завдання або портфоліо.

7. Критерії та форми контролю

Оцінювання — за 100-бальною шкалою (національна шкала / ЄКТС). Підсумковий бал = поточний контроль (25 %) + модульний контроль (20 %) + курсова робота (15 %) + екзамен (40 %).

Академічна доброчесність: самостійне виконання завдань; коректне посилання на джерела; позначення частин тексту, згенерованих засобами ШІ. Порушення академічної доброчесності не толеруються.

8. Зміст тем

Тема 1. Парадигми програмування. Вступ до ООП

Поняття парадигми програмування. Імперативне, декларативне, процедурне, структурне та модульне програмування. Причини появи об’єктно-орієнтованого підходу. Клас, об’єкт, стан, поведінка та ідентичність. Абстрагування, інкапсуляція, успадкування і поліморфізм. Переваги, обмеження та сфери застосування ООП.

Тема 2. Мова C# і платформа .NET

Історичні передумови створення C# і .NET. Місце C# серед сучасних мов програмування. Складові платформи .NET: SDK, середовище виконання, базова бібліотека класів і CLR. Компілятор Roslyn, проміжний код IL, JIT-компіляція, керований і некерований код. Автоматичне керування пам’яттю та Garbage Collector. Структура solution, проєкту й файла .csproj. Типи застосунків .NET та інструменти Visual Studio і .NET CLI.

Тема 3. Основи мови C#

Структура програми C# та інструкції верхнього рівня. Змінні, константи, вбудовані типи даних, перетворення типів, арифметичні й логічні операції. Консольне введення і виведення, інтерполяція та форматування. Умовні конструкції if і switch. Цикли for, while, do while та foreach. Основи масивів і рядків. Перевірка введених даних за допомогою TryParse. Компіляція, запуск і налагодження програми.

Тема 4. Класи, об’єкти, структури та простори імен

Клас як опис нового типу, об’єкт як екземпляр класу. Поля, властивості та методи. Створення об’єктів. Конструктори, ключове слово this, перевантаження і ланцюжки конструкторів, ініціалізатори об’єктів. Статичні й нестатичні члени. Структури та відмінності від класів. Простори імен, директива using, системні простори імен і бібліотеки класів.

Тема 5. Методи та декомпозиція програми

Оголошення і виклик методів. Статичні та екземплярні методи. Тип повернення і оператор return. Параметри методів, передавання за значенням і за посиланням, ref, out та in. Іменовані й необов’язкові параметри, params. Перевантаження методів. Область видимості локальних змінних. Декомпозиція задачі на невеликі методи з чіткою відповідальністю.

Тема 6. Успадкування

Базові та похідні класи. Відношення «є різновидом». Доступ protected. Ключові слова base і sealed. Успадкування полів, властивостей і методів. Конструктори у похідних класах та порядок їх виклику. Перетворення посилань між базовим і похідним типами, оператори is та as.

Доцільність успадкування і переваги композиції.

Тема 7. Віртуальні методи, абстрактні класи та поліморфізм

Статичний і динамічний поліморфізм. Віртуальні методи та властивості, ключові слова virtual, override і base. Приховування членів за допомогою new та відмінність від перевизначення. Заборона перевизначення. Абстрактні класи, абстрактні члени й неможливість створення їх екземплярів. Поліморфний виклик через посилання базового типу.

Тема 8. Інтерфейси

Інтерфейс як контракт поведінки. Оголошення інтерфейсів і реалізація їх класами та структурами. Реалізація кількох інтерфейсів, успадкування інтерфейсів, явна реалізація. Інтерфейси у базових і похідних класах. Використання інтерфейсів для слабкого зв’язування компонентів. Огляд стандартних інтерфейсів IComparable, IEquatable та IEnumerable.

Тема 9. Типи даних, узагальнення та колекції

Типи значень і посилальні типи, null та nullable-типи. Узагальнені класи й методи, параметри типу та обмеження узагальнень. Переваги типобезпечності й повторного використання коду. Колекції List<T>, Dictionary<TKey,TValue>, Stack<T>, Queue<T> та HashSet<T>. Вибір колекції відповідно до задачі. Інтерфейси IEnumerable<T> та IEnumerator<T>, перебір даних.

Тема 10. Делегати, лямбда-вирази та події

Делегат як типобезпечне посилання на метод. Створення, призначення і виклик делегатів, групові делегати. Узагальнені делегати Action, Func і Predicate. Анонімні методи й лямбда-вирази, захоплення змінних. Передавання поведінки як параметра. Події та модель «видавець — підписник», оголошення, підписка й відписка. EventHandler і передавання даних події.

Тема 11. LINQ і декларативна обробка даних

Призначення LINQ і декларативний стиль опрацювання даних. Джерела даних IEnumerable<T>. Синтаксис методів і запитів. Фільтрування, проєкція, сортування, групування, об’єднання та агрегування. Методи Where, Select, OrderBy, GroupBy, Join, Any, All, Count, Sum і Average. Відкладене виконання, матеріалізація результатів і типові помилки під час побудови запитів.

Тема 12. Графічний інтерфейс користувача в .NET

Принципи подієво-керованого графічного застосунку. Огляд Windows Forms, WPF, .NET MAUI та Avalonia UI; вибір технології залежно від операційної системи й мети проєкту. Вікно, елементи керування, властивості, події та обробники. Компонування інтерфейсу, введення, відображення й валідація даних. Відокремлення інтерфейсу від предметної логіки. Основи прив’язування даних.

Тема 13. Мережеві запити, JSON та асинхронне програмування

Робота з мережею за допомогою HttpClient. URL, HTTP-запит і відповідь, коди стану та веб-API. Формат JSON і дослідження відповіді сервера. Серіалізація та десеріалізація засобами System.Text.Json. Асинхронні операції, Task, ключові слова async і await, скасування операцій. Відмінність асинхронності, багатопотоковості й паралельного виконання. Отримання даних без блокування графічного інтерфейсу.

Тема 14. Обробка винятків, файли та потоки даних

Синтаксичні, логічні помилки та помилки виконання. Механізм винятків .NET, конструкції try, catch і finally, оператор throw, фільтри та власні типи винятків. Принципи коректної обробки помилок. Простір імен System.IO, файли й каталоги, потоки Stream, StreamReader і StreamWriter. Читання та запис тексту, JSON і CSV, асинхронні файлові операції, using та звільнення ресурсів.

Тема 15. SQLite, Entity Framework Core та CRUD

Реляційні бази даних і призначення SQLite. Пакети NuGet та підключення Entity Framework Core. Сутності, первинні ключі, DbContext і DbSet. Налаштування зв’язку з базою даних, створення схеми та міграції. Операції CRUD: створення, читання, оновлення і видалення. Асинхронні запити та LINQ to Entities. Відображення даних у графічному інтерфейсі, валідація введення й розділення інтерфейсу, бізнес-логіки та доступу до даних.

9. Курсова робота

Курсова робота є індивідуальним програмним проєктом мовою C#, у якому здобувач застосовує принципи об’єктно-орієнтованого проєктування для розв’язання прикладної задачі відповідно до обраної або призначеної теми. На її виконання відводиться 30 годин самостійної роботи.

Робота передбачає аналіз предметної області та вимог, проєктування класів і зв’язків між ними, реалізацію працездатного застосунку, перевірку його роботи та підготовку пояснювальної записки. Програма повинна мати зрозумілу структуру, коректно використовувати інкапсуляцію, композицію або успадкування, поліморфізм чи інтерфейси там, де це обґрунтовано, обробляти помилкові дані та зберігати потрібну інформацію. Конкретний тип інтерфейсу і спосіб збереження даних визначаються темою та погоджуються з керівником.

Пояснювальну записку оформлюють за чинними методичними вказівками кафедри. Вона має містити постановку задачі, аналіз предметної області, опис прийнятих рішень і структури програми, опис реалізації, приклади роботи та результати перевірки. Код і використані джерела подають із дотриманням академічної доброчесності; використання інструментів штучного інтелекту декларується.

Захист передбачає демонстрацію програми, пояснення архітектури та ключових фрагментів власного коду, відповіді на запитання і, за потреби, внесення невеликої зміни до програми. Оцінюються відповідність завданню, коректність роботи, обґрунтованість об’єктної моделі, якість коду й оформлення, самостійність виконання та здатність пояснити прийняті рішення.

10. Самостійна робота

Самостійна робота спрямована на закріплення матеріалу лекцій, підготовку до лабораторних занять, модульного та підсумкового контролю. Загальний обсяг становить 135 годин для денної та 196 годин для заочної форми навчання. Із них по 30 годин відведено на курсову роботу; безпосередньо за темами дисципліни — 105 і 166 годин відповідно.

Основні види самостійної роботи:

- опрацювання конспектів лекцій, рекомендованої літератури й офіційної документації C# та .NET;
- повторення синтаксису C# і виконання невеликих тренувальних прикладів;
- підготовка до лабораторних робіт, завершення й налагодження програм, перевірка їх на звичайних, граничних і помилкових вхідних даних;
- аналіз прикладів використання класів, інкапсуляції, успадкування, поліморфізму, інтерфейсів, узагальнень, колекцій, делегатів, подій і LINQ;
- самостійне опрацювання документації з технологій графічного інтерфейсу .NET, HttpClient, JSON, async/await, System.IO, SQLite та Entity Framework Core;
- пошук і критичне оцінювання технічної інформації, ведення коротких нотаток про використані джерела та прийняті програмні рішення;
- підготовка до усного захисту лабораторних робіт: пояснення структури програми, призначення класів і методів, причин вибору алгоритмів та можливих змін коду;
- виконання курсової роботи відповідно до окремих вимог розділу «Курсова робота».

Самостійна робота виконується протягом семестру відповідно до послідовності тем. Формами контролю є перевірка підготовки до занять, працездатності та самостійності виконання програмного коду, усний захист лабораторних робіт, модульні контрольні роботи, іспит і захист курсової роботи. Використання засобів штучного інтелекту допускається лише відповідно до загальних вимог силабуса, з обов’язковою перевіркою, розумінням і декларуванням їх застосування.

11. Загальні вимоги

Освітній процес регулюється академічними правилами ЛНТУ. Здобувачі освіти зобов’язані ознайомитися з вимогами, визначеними Кодексом академічної доброчесності ЛНТУ та Політикою етичного використання штучного інтелекту в ЛНТУ, і дотримуватися принципів

академічної доброчесності та етичного використання штучного інтелекту в освітній діяльності.

Академічна доброчесність. Учасники освітнього процесу зобов’язані дотримуватися Закону України «Про академічну доброчесність» від 18.12.2025 № 4742-IX та внутрішніх нормативних документів ЛНТУ. Не допускаються академічний плагіат, фабрикація, фальсифікація, привласнення авторства, несамостійне виконання завдань, а також недоброчесне використання результатів, згенерованих штучним інтелектом.

Використання штучного інтелекту. Інструменти штучного інтелекту можуть використовуватися для пошуку ідей, пояснення методів, попереднього структурування, мовного редагування та допоміжного аналізу неконфіденційних даних, якщо конкретне завдання не передбачає повністю самостійного виконання. Здобувач освіти має зазначати використаний інструмент, мету та етап його застосування, перевіряти факти й розрахунки та самостійно обґрунтовувати висновки.

12. Література

I. Навчально-методичні матеріали та розробки

Праці автора силабуса:

1. Ящук А. А. Об’єктно-орієнтоване програмування : конспект лекцій. Луцьк : ЛНТУ, 2024. 44 с.

2. Ящук А. А. Об’єктно-орієнтоване програмування : методичні вказівки до виконання лабораторних робіт. Луцьк : ЛНТУ, 2024. 100 с.

3. Ящук А. А. Об’єктно-орієнтоване програмування : методичні вказівки до виконання курсової роботи. Луцьк : ЛНТУ, 2024. 36 с.

4. Ящук А. А. Об’єктно-орієнтоване програмування : методичні вказівки до самостійної роботи. Луцьк : ЛНТУ, 2024. 18 с.

Праці інших авторів:

5. Price M. J. C# 14 and .NET 10 – Modern Cross-Platform Development Fundamentals. 10th ed. Birmingham : Packt Publishing, 2025. 828 p.

6. Stellman A., Greene J. Head First C#. 5th ed. Sebastopol : O’Reilly Media, 2024. 838 p.

7. Albahari J. C# 12 in a Nutshell. Sebastopol : O’Reilly Media, 2023. 1086 p.

8. Whitaker R. B. The C# Player’s Guide. 5th ed. Starbound Software, 2022. 495 p.

II. Наукові праці

Праці автора силабуса:

9. Horshkov V., Lishchyna N., Yashchuk A., Surynovych O., Lishchyna V. A modular monolith architecture for high-availability electric vehicle charging station management software. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska. 2026. Vol. 16, no. 3. P. 95–102.

10. Surynovych O., Lishchyna N., Yashchuk A., Povstiana Y., Lishchyna V., Kukharchuk D. A Web-Based Approach for Assessing CO2 Emissions during the Russo-Ukrainian War. 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT). Athens, 2024. P. 1–6. DOI: 10.1109/DESSERT65323.2024.11122227.

11. Kozubtsova L., Ali R. H. M., Kozubtsov I., Lishchyna V., Yashchuk A., Hassan N. B., Lukashenka V. Approach to Risk Management Based on the Assessment of the Cost of Quality of Implementation of Cybersecurity Measures of the Organization. 2024 35th Conference of Open Innovations Association (FRUCT). Tampere, 2024. P. 399–406.

III. Інформаційні ресурси

12. Microsoft. Документація C#. URL: https://learn.microsoft.com/uk-ua/dotnet/csharp/ (дата звернення: 25.08.2026).

13. Microsoft. Документація .NET. URL: https://learn.microsoft.com/uk-ua/dotnet/ (дата звернення: 25.08.2026).

14. Microsoft. Документація Visual Studio. URL: https://learn.microsoft.com/uk-ua/visualstudio/ (дата звернення: 25.08.2026).

15. Microsoft. Документація Entity Framework Core. URL: https://learn.microsoft.com/uk-ua/ef/core/ (дата звернення: 25.08.2026).

16. Microsoft. Документація NuGet. URL: https://learn.microsoft.com/uk-ua/nuget/ (дата звернення: 25.06.2026).

17. Avalonia UI. Documentation. URL: https://docs.avaloniaui.net/ (дата звернення: 25.08.2026).

18. SQLite. Documentation. URL: https://www.sqlite.org/docs.html (дата звернення: 25.08.2026).