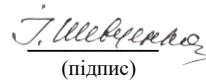


Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

І.В. Шевченко
(ініціали та прізвище)

« 29 » серпня 2025 р.

СИЛАБУС
ОБОВ'ЯЗКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

(назва навчальної дисципліни)

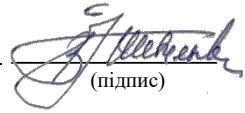
СТУПІНЬ ВИЩОЇ ОСВІТИ	Бакалавр
ГАЛУЗЬ ЗНАНЬ	12 (F) Інформаційні технології
СПЕЦІАЛЬНІСТЬ	121 (F2) Інженерія програмного забезпечення
ОСВІТНЯ ПРОГРАМА	Інженерія програмного забезпечення

Рівень вищої освіти: *перший (бакалаврський)*

Силабус введено в дію з 01.09.2025 року

Харків – 2025 р.

Розробник: Шевченко І.В., доц. каф. 603, канд. техн. наук., доц.
(прізвище та ініціали, посада, наукова ступінь та вчене звання)

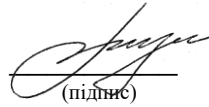

(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії
програмного забезпечення

(назва кафедри)

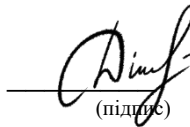
Протокол № 1 від «29» серпня 2025 р.

Завідувач кафедри д-р техн. наук., проф.
(науковий ступінь і вчене звання)


(підпис)

І.Б. Туркін
(ініціали та прізвище)

Погоджено з представником здобувачів освіти:
Представник студентського самоврядування


(підпис)

Дикун Д.В.
(ініціали та прізвище)

1. Загальна інформація про викладача



ПІБ: Шевченко Ілона Володимирівна

Посада: доцент кафедри інженерії програмного забезпечення

Науковий ступінь: кандидат технічних наук

Вчене звання: доцент

Перелік дисциплін, які викладає:

«Об'єктно-орієнтоване програмування»

«Алгоритми систем підтримки прийняття рішень»

«NoSQL бази даних»

Напрями наукових досліджень:

Decision Support System, NoSQL databases, Big Data, Machine learning

Контактна інформація:

i.shevchenko@khai.edu

@IlonaShev

2. Опис навчальної дисципліни

Форма здобуття освіти	<i>Денна</i>
Семестр	3 семестр
Мова викладання	Українська
Тип дисципліни	<i>обов'язкова</i>
Обсяг дисципліни: кредити ЄКТС/ кількість годин	<i>денна</i> : 5 кредитів ЄКТС / 150 годин (64 аудиторних, з яких: лекції – 32, практичні – 32; СРЗ – 86).
Види навчальної діяльності	Лекції, практичні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	«Основи програмування», «Програмування мовою С#»

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета вивчення – надати студентам знання і сформувати навички щодо розроблення програмного забезпечення за допомогою об'єктно-орієнтованої парадигми програмування.

Завдання – вивчення теоретичного матеріалу, відпрацювання на практичних заняттях та виконання розрахункової роботи (у формі проєкта) щодо використання об'єктно-орієнтованої парадигми програмування при розробленні програмного забезпечення.

Компетентності, які набуваються:

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.

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде мати:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК13. Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь-яких інших проявів не доброчесності.

Фахові компетентності (ФК)

Після закінчення цієї програми здобувач освіти буде мати:

ФК02. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК03. Здатність розробляти архітектури, модулі та компоненти програмних систем.

ФК07. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання (ПРН):

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

ПРН06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

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

ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

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

3. Зміст навчальної дисципліни

Модуль 1. Об'єктно-орієнтоване програмування: початковий рівень

Тема 1. Класи та об'єкти. Абстракція.

Особливості ООП як парадигми програмування. Поняття «клас» і «об'єкт класу». Принцип абстракції. Характеристики та поведінка об'єкту класу. Class: поля, методи, модифікатори доступу. Типи значень і типи посилань. Клас як представник типу посилань. Розроблення власного класу і створення масиву (колекції) об'єктів власного класу.

Тема лекції: «Класи та об'єкти. Абстракція».

Тема практичного заняття: «Класи, об'єкти».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 2. Інкапсуляція. Властивості.

Принцип інкапсуляції. Поняття «властивість». Особливості реалізації властивостей. Обчислювальні властивості. Автоматичні властивості. Властивості і модифікатори доступу. Скорочений запис властивостей. Властивості і Exceptions.

Тема лекції: «Інкапсуляція, властивості».

Тема практичного заняття: «Інкапсуляція, властивості».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 3. Конструктор класу. Перевантаження конструктору класу. Ініціалізатори. Деконструктори. Перевантаження методів класу.

Поняття «конструктор класу». Конструктор за замовчуванням. Власний конструктор. Передача інформації конструктору. Перевантаження конструктору класу. Використання ініціалізаторів при створенні об'єктів класу. Поняття «деконструктор класу». Перевантаження методів класу. Поняття «сигнатура метода».

Тема лекції: «Конструктор класу. Перевантаження конструктору класу. Ініціалізатори. Деконструктори. Перевантаження методів класу».

Тема практичного заняття: «Конструктори, перевантаження».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 4. Статичні: поля, властивості, методи, конструктори, класи.

Ключове слово «static». Особливості статичних полів, властивостей, методів, конструкторів, класів. Статичні класи Console, Convert, Math. Статичні методи класу String. Реалізація статичних методів Parse і TryParse для власного класу.

Тема лекції: «Статичні: поля, властивості, методи, конструктори, класи».

Тема практичного заняття: «Статичні поля, властивості, методи».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 5. Модульне тестування (Unit testing). Test-Driven-Development (TDD).

Що таке модульне тестування? Фреймворки для проведення модульного тестування: xUnit, MSTest, NUnit. Особливості застосування фреймворку MSTest для проведення модульного тестування: атрибути [TestClass] і [TestMethod], AAA-шаблон для написання тестового методу, клас Assert і його методи.

Test-Driven-Development: поняття, особливості.

Тема лекції: «Модульне тестування (Unit testing). Test-Driven-Development (TDD)».

Тема практичного заняття: «Модульне тестування (unit testing)».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 6. Узагальнені типи (Generics). Колекції (Collections).

Поняття «упакування (boxing)» і «розпакування (unboxing)». Поняття «узагальнений тип (generic)». Поняття «узагальнений клас». Особливості використання статичних полів в узагальненому класі. Використання декількох універсальних параметрів в узагальненому класі. Узагальнений метод.

Колекції (Collections). List<T>: створення типизованого списку та використання його методів. Stack<T>: створення типизованого стеку та використання його методів. Queue<T>: створення типизованої черги та використання її методів. Dictionary<K, V>: створення типизованого словника та використання його методів.

Тема лекції: «Узагальнені типи (Generics). Колекції (Collections)».

Тема практичного заняття: «Generics. Collections».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Тема 7. Збереження та завантаження об'єктів класу (Saving and loading of objects). Серіалізація (Serialization) / Десеріалізація (Deserialization)

*Поняття «серіалізації» (serialization) і «десеріалізації» (deserialization) об'єктів класу. Особливості форматів файлів *.csv та *.json. Реалізація збереження списку об'єктів класу у файл *.csv та *.json. Реалізація завантаження списку об'єктів класу з файлів *.csv (*.txt) та *.json.*

Тема лекції: «Збереження та завантаження об'єктів класу (Saving and loading of objects). Серіалізація (Serialization) / Десеріалізація (Deserialization)».

Тема практичного заняття: «Saving and loading of objects. Serialization/Deserialization».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, завершення виконання практичної роботи, оформлення звіту та підготовка до захисту практичної роботи.

Модульний контроль 1

Проходження комп'ютерного тестування і кодингу з часовим лімітом за рахунок часу на самостійну роботу.

Модуль 2. Об'єктно-орієнтоване програмування: середній рівень

Тема 8. Відношення між класами: асоціація, агрегація, композиція

Поняття «асоціація», «агрегація», «композиція» як відношення між класами. Особливості реалізації асоціації, агрегації, композиції.

Тема лекції: «Відношення між класами: асоціація, агрегація, композиція».

Тема практичного заняття: «Відношення між класами».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part A).

Тема 9. Відношення між класами: реалізація. Інтерфейси

Особливості відношення «реалізація». Поняття «інтерфейс». Реалізація інтерфейсів. Посилання на інтерфейси. Інтерфейси і модифікатори доступу. Множинна реалізація інтерфейсів. Реалізація інтерфейсів за замовчуванням. Неявна і явна реалізація інтерфейсів.

.NET-інтерфейси: `ICloneable`, `Comparable<T>`, `IEnumerable`, `IEnumerator`. Їх особливості і приклади використання.

Тема лекції: «Відношення між класами: реалізація. Інтерфейси»;

Тема практичного заняття: «Інтерфейси»;

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part A).

Тема 10. Відношення між класами: спадкування (Inheritance)

Поняття «успадкування (inheritance)». Відмінність успадкування від реалізації інтерфейсу. Створення класів-нащадків. Правила виклику конструкторів базових класів. Рівні доступу при спадкуванні.

Тема лекції: «Відношення між класами: спадкування (Inheritance)».

Тема практичного заняття: «Спадкування класів».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part B).

Тема 11. Віртуальні методи. Поліморфізм.

Перевизначення (overriding) методів у класах-нащадках. Віртуальні (virtual) методи. Використання базових (base) методів. Заміна (replacement) методів у класах-нащадках. Sealed-методи. Sealed-класи.

Принцип поліморфізму (polymorphism). Варіанти реалізації поліморфізму.

Клас `System.Object` – загальний базовий клас. Методи класу `System.Object`.

Тема лекції: «Віртуальні методи. Поліморфізм».

Тема практичного заняття: «Віртуальні методи. Поліморфізм».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part B).

Тема 12. Абстрактні (abstract) методи та класи

Поняття «абстрактний (abstract) метод». Поняття «абстрактний (abstract) клас». Схожість і відмінність абстрактного класу і інтерфейсу. Обмеження щодо абстрактних членів класу. Абстрактні члени як частина поліморфного інтерфейсу. Відмова від реалізації абстрактних членів класу.

Реалізація інтерфейсів у базових та похідних класах. Зміна реалізації інтерфейсів у похідних класах. Снадкування інтерфейсів.

Тема лекції: «Абстрактні (abstract) методи та класи».

Тема практичного заняття: «Абстрактні методи та класи».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part B).

Тема 13. Перетворення типів: висхідні перетворення (upcasting) і низхідні перетворення (downcasting)

Особливості висхідного перетворення типів (upcasting). Приклади висхідного перетворення типів. Особливості низхідного перетворення типів (downcasting). Оператор «as». Оператор «is». Приклади низхідного перетворення типів.

Тема лекції: «Перетворення типів: висхідні перетворення (upcasting) і низхідні перетворення (downcasting)».

Самостійна робота здобувача освіти: повторення лекційного матеріалу.

Модульний контроль 2

Проходження комп'ютерного тестування і кодингу з часовим лімітом за рахунок часу на самостійну роботу.

Модуль 3. Об'єктно-орієнтоване програмування: поглиблений рівень

Тема 14. Делегати (delegates)

Поняття «делегат (delegate)». Визначення делегата. Робота з делегатами. Відповідність методів делегату. Додавання методів у делегат. Видалення методів з делегата. Об'єднання делегатів. Повернення значення делегатом. Узагальнені делегати. Делегати як параметри методів. Повернення делегатів із методу. Приклади використання делегатів.

.NET-делегати: Action, Predicate, Func. Їх особливості та приклади використання.

Тема лекції: «Делегати (delegates)».

Тема практичного заняття: «Делегати».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part C).

Тема 15. Події (events)

Поняття «події (events)». Визначення та виклик подій. Додавання обробника подій. Передача даних подій. Приклади використання подій.

Тема лекції: «Події (events)».

Тема практичного заняття: Події.

Самостійна робота здобувача освіти: повторення лекційного матеріалу, виконання Project (part C), підготовка до захисту Project.

Тема 16. Принципи SOLID

SOLID: «S» – Single Responsibility Principle, «O» – Open/Close Principle, «L» – Liskov Substitution Principle, «I» – Interface Segregation Principle, «D» –Dependency Inversion Principle.

Тема лекції: «Принципи SOLID».

Самостійна робота здобувача освіти: повторення лекційного матеріалу.

Модульний контроль 3

Проходження комп'ютерного тестування за рахунок часу на самостійну роботу.

Модуль 4. Об'єктно-орієнтоване програмування: додаткові можливості

Тема 17. Анонімні методи. Лямбда-вирази. Методи розширення.

Поняття «анонімний метод». Визначення анонімних методів. Приклади використання анонімних методів.

Поняття «лямбда-виразу». Визначення лямбда-виразу. Лямбда-вираз як аргумент метода. Лямбда-вираз як результат метода. Приклади використання лямбда-виразів.

Поняття «метод розширення (extention method)». Визначення методів розширення. Приклади використання методів розширення.

Тема лекції: «Анонімні методи. Лямбда-вирази. Методи розширення».

Самостійна робота здобувача освіти: повторення лекційного матеріалу.

Тема 18. Мова запитів до джерела даних LINQ

Поняття «Language-Integrated Query». Різновиди LINQ. LINQ to Objects: запити до масивів та колекцій. LINQ toEntities: запити до баз даних через технологію EntityFramework. LINQ toXML: запит до файлів XML. LINQ toDataSet: запит до об'єкту DataSet. ParallelLINQ (PLINQ): паралельні запити.

Особливості роботи з LINQ to Objects. Оператори запитів LINQ. Методи розширень LINQ. Проекція даних. Фільтрація даних. Сортування даних. Перевірка наявності і отримання елементів. Отримання частини колекції.

Тема лекції: «Мова запитів до джерела даних LINQ».

Тема практичного заняття: «LINQ».

Самостійна робота здобувача освіти: повторення лекційного матеріалу, комп'ютерне тестування (дистанційна форма).

Тема 19. Оператори. Індексатори. Часткові класи і методи. Анонімні типи. Кортежі.

Визначення оператора класу. Особливості розроблення операторів класу. Перевантаження оператору класу. Явні і неявні оператори перетворення типів. Приклади використання операторів класу.

Визначення індексатора класу. Застосування декількох параметрів індексування. Перевантаження індексаторів. Приклади використання індексаторів.

Визначення часткового (partial) класу і методу. Застосування часткових класів і методів. Приклади використання часткових методів і класів.

Визначення анонімного типу. Приклади використання анонімних типів.

Визначення кортежу. Особливості використання кортежів. Кортеж як параметр методу. Кортеж як результат методу. Приклади використання кортежів.

Тема лекції: «Оператори. Індексатори. Часткові класи і методи. Анонімні типи. Кортежі».

Самостійна робота здобувача освіти: повторення лекційного матеріалу.

Модульний контроль 4

Пройходження комп'ютерного тестування за рахунок часу на самостійну роботу.

5. Індивідуальні завдання

Виконання індивідуального завдання (*Project*) за темою «Побудова моделі предметної області у вигляді діаграми класів та TDD-розроблення програмного забезпечення».

Part A: Побудова моделі предметної області у вигляді діаграми класів, програмна реалізація каркасів класів і розроблення unit-tests.

Part B: Розвиток діаграми класів: успадкування, абстрактні класи. Програмна реалізація методів класів.

Part C: Програмна реалізація програмного прототипу.

6. Методи навчання

Словесні, наочні, практичні.

7. Методи контролю

Поточний контроль (теоретичні опитування під час захисту практичних робіт), модульний контроль (комп'ютерне дистанційне тестування) та підсумковий (семестровий) контроль (іспит).

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять	Сумарна кількість балів
Модуль 1			
Виконання і захист практичних робіт №1-7	0..5	7	0..35
Модульний контроль 1	0..5	1	0..5
Кодинг 1	0..10	1	0..10
Модуль 2			
Виконання Project (Part A)	0..10	1	0..10
Виконання Project (Part B)	0..10	1	0..10
Модульний контроль 2	0..5	1	0..5
Кодинг 2	0..10	1	0..10
Модуль 3			
Виконання Project (Part C)	0..5	1	0..5
Модульний контроль 3	0..5	1	0..5
Модуль 3			
Модульний контроль 4	0..5	1	0..5
Усього за семестр			0..100

Семестровий контроль (іспит) проводиться у разі відмови здобувача від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту здобувач має можливість отримати максимум 100 балів.

Для отримання оцінки ≤ 74 балів іспит проводиться у формі комп'ютерного тестування. Тест складається з 37 питань (за правильну відповідь на одне питання здобувач отримує 2 бали). Підсумкова оцінка за іспит розраховується за формулою: Бал за іспит = 0 + набрані тестові бали.

Для отримання оцінки ≥ 75 балів іспит проводиться у форматі кодингу з часовими лімітом (час на виконання – 2 години). За кодинг студент може набрати максимально 40 балів. Підсумкова оцінка за іспит розраховується за формулою: Бал за іспит = 60 + бали за кодинг.

Таблиця 8.2 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Задовільно (60-74). Здати всі практичні роботи (зі значною корекцією тексту програм практичних робіт викладачем), захистити розрахункову роботу (зі значною корекцією тексту програми розрахункової роботи викладачем), мати необхідних мінімум знань за всіма темами та мінімум вмінь щодо застосування отриманих знань.

Добре (75-89). Здати всі практичні роботи (з мінімальною корекцією тексту програм практичних робіт викладачем), захистити розрахункову роботу (з мінімальною корекцією тексту програми розрахункової роботи викладачем), знати всі теми та уміти застосовувати їх.

Відмінно (90-100). Здати всі практичні роботи (без корекції тексту програми практичних робіт викладачем), захистити розрахункову роботу (без корекції тексту програми розрахункової роботи викладачем), досконально знати всі теми та уміти застосовувати їх.

9. Політика навчального курсу

Відвідування занять. Курс передбачає обов'язкове відвідування лекційних і практичних занять. Здобувачі освіти, які з певних причин не можуть регулярно відвідувати заняття, **зобов'язані** переглядати записи лекцій (посилання на записи викладаються у системі Mentor) та самостійно опановувати теоретичний матеріал, щодо пропуску практичних робіт, то здобувач самостійно ознайомлюється з інструкцією до практичної роботи, виконує її, оформлює звіт і потім захищає роботу на парі; допускається захист кількох робіт одночасно, але не більше трьох практичних робіт за раз.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>).

Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем.

Використання ІІІ. Дозволяється застосовувати інструменти генеративного ІІІ (наприклад, ChatGPT, Copilot тощо) для пошуку довідкової інформації, пояснень синтаксису, прикладів команд або перевірки власних рішень. Не допускається повне копіювання згенерованих рішень без їх розуміння та адаптації, оскільки це не сприяє формуванню практичних навичок. Усі практичні роботи здобувач повинен уміти самостійно пояснити. У деяких практичних роботах для певних задач викладачем буде навіть рекомендовано використання ІІІ.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu/ua/university/normativna-baza/ustanovchi-dokumenty/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

Розроблений дистанційний курс дисципліни знаходиться за посиланням: <https://mentor.khai.edu/course/view?id=7388>

Дистанційний курс містить посилання на записи лекційних занять, посилання на навчальний посібник до практичних робіт, рекомендовану літературу.

11. Рекомендована література

1. Презентації до лекційних занять (див. курс в Mentor).
2. Шевченко, І. В. Об'єктно-орієнтоване програмування (мова C#) [Електронний ресурс] : навч. посіб. до практ. робіт / І. В. Шевченко, П. О. Лучшев. – Харків : Нац. аерокосм. ун-т «Харків. авіац. ін-т», 2025. – 99 с.
3. Rob Miles. C# programming. Yellow Book, 2019.
4. Ian Griffiths. Programming C# 10: Build Cloud, Web, and Desktop Applications, 2022.

5. Andrew Troelsen. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming, 2022.
6. Jon Skeet. C# in Depth, 2019.

12. Інформаційні ресурси

1. C# programming guide. – Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/>
2. C# Tutorial. – Режим доступу:
<https://www.w3schools.com/cs/index.php>
3. Learn C# Programming. – Режим доступу:
<https://www.programiz.com/csharp-programming>
4. Learn C# Programming. – Режим доступу:
<https://www.tutorialsteacher.com/csharp>
5. Learn Object-Oriented Programming with C#. – Режим доступу:
<https://www.tutorialsteacher.com/csharp/oop>
6. Language Integrated Query (LINQ) (C#). – Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>
7. Learn LINQ (Language-Integrated Query). – Режим доступу:
<https://www.tutorialsteacher.com/linq>