



**Приватний вищий навчальний заклад
«Буковинський університет»
Факультет інформаційних технологій та економіки
Кафедра комп'ютерних систем і технологій**

СХВАЛЕНО
на засіданні науково-методичної
ради факультету
протокол № 1 від 26 серпня 2025 р.

ЗАТВЕРДЖУЮ
Декан факультету ІТЕ
_____/ Тетяна ШТЕРМА/
«___» _____ 2025 р.

СИЛАБУС
обов'язкової навчальної дисципліни
«Об'єктно-орієнтоване програмування»

Освітньо-професійна програма:	<u>Комп'ютерні науки</u>
Спеціальність:	<u>Комп'ютерні науки</u>
Галузь знань:	<u>Інформаційні технології</u>
Рівень вищої освіти:	<u>перший (бакалаврський)</u>
Факультет:	<u>Інформаційних технологій та економіки</u>
Мова навчання:	<u>українська</u>
Розробник:	Осадчук Сергій Іванович – викладач
Профайл викладача:	https://bukuniver.edu.ua/university/faculties-and-departments/ite-faculty/department-of-computer-systems-and-technologies/#
E-mail:	serhiosadchuk@bukuniver.edu.ua
Консультації:	четвер з 10.00 до 16.00.

1. Анотація (призначення навчальної дисципліни).

Освітній компонент «Об'єктно-орієнтоване програмування» є складовою навчального плану підготовки фахівців першого (бакалаврського) рівня вищої освіти галузі знань **Інформаційні технології спеціальності Комп'ютерні науки**, яка побудована відповідно до вимог Європейської кредитної трансферно-накопичувальної системи (ECTS) і містить 4 кредити. Форма підсумкового контролю – екзамен.

Предметом вивчення курсу є парадигма об'єктно-орієнтованого програмування, її принципи (інкапсуляція, успадкування, поліморфізм) та їх практична реалізація мовою C# на платформі .NET. Курс спрямований на формування навичок проєктування та розробки гнучких, розширюваних програмних систем. Значну увагу приділено роботі з класами, інтерфейсами, винятками, колекціями, делегатами та подіями, а також основам Unit-тестування та серіалізації даних.

2. Мета та завдання вивчення дисципліни.

Мета курсу – сформувати у здобувачів вищої освіти систему теоретичних знань та практичних навичок щодо об'єктно-орієнтованого аналізу, проєктування та програмування для створення якісного програмного забезпечення.

Завдання дисципліни :

- забезпечити розуміння базових принципів ООП та їх реалізації в мові C#;
- сформувати навички створення власних типів даних (класів, структур, перелічень);
- навчити використовувати механізми успадкування та поліморфізму для повторного використання коду;
- опанувати роботу зі стандартними бібліотеками .NET (колекції, файловий ввід/вивід, LINQ);
- навчити обробляти виняткові ситуації та проводити модульне тестування (Unit Testing).

Основні знання та вміння, яких набуває студент після опанування цієї дисципліни

Основні знання:

- синтаксису та семантики мови C#;
- принципів роботи платформи .NET Framework та управління пам'яттю;
- правил створення та використання класів, об'єктів, інтерфейсів;
- механізмів обробки винятків та роботи з потоками даних;
- основних шаблонів проєктування та принципів SOLID.

Основні вміння:

- проєктувати ієрархію класів для моделювання предметної області;
- реалізовувати інтерфейси та використовувати поліморфізм ;
- використовувати механізм LINQ для роботи з даними ;
- розробляти Unit-тести для перевірки коректності коду ;
- здійснювати серіалізацію та десеріалізацію об'єктів (XML, JSON, Binary).

3. Пререквізити:

- ОК4 Дискретна математика;
- ОК16 Алгоритмізація та програмування;
- ОК20 Алгоритми і структури даних.

4. Компетентності та результати навчання.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються компетентності:

Інтегральна компетентність

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов і вимог.

Загальні компетентності (ЗК)

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Здатність вчитися і оволодівати сучасними знаннями.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові, предметні) компетентності

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм

програмування: узагальнення, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК17. Здатність здійснювати збір, оброблення та аналіз структурованих та слабоструктурованих даних, а також проектувати й впроваджувати інтелектуальні інформаційні системи та системи підтримки прийняття рішень із використанням сучасних програмних інструментів, методів аналізу даних і технологій штучного інтелекту у прикладних галузях діяльності.

Програмні результати навчання

РН5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

РН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

РН20. Використовувати сучасні програмні інструменти, методи аналізу даних і технології штучного інтелекту для збору, оброблення та аналізу структурованих і слабоструктурованих даних під час розв'язання прикладних задач.

5. Зміст навчальної дисципліни

Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми та інкапсуляція

Тема 1. Вступ до об'єктно-орієнтованого програмування: концепція класів та об'єктів, архітектура .NET та синтаксис C#.

Тема 2. Інкапсуляція даних: модифікатори доступу, властивості (get/set) та автоматичні властивості.

Тема 3. Конструктори та життєвий цикл об'єкта: ініціалізація стану, перевантаження методів та конструкторів.

Тема 4. Керування пам'яттю: значущі (Value types) та посилальні (Reference types) типи даних, робота зі стеком та купою.

Тема 5. Статичні члени класів: використання ключового слова static, статичні методи та класи-утиліти.

Тема 6. Відносини між об'єктами: реалізація асоціації, агрегації та композиції у програмному коді.

Змістовий модуль 2. Механізми повторного використання та розширення систем

Тема 7. Успадкування: створення ієрархії класів, базові та похідні класи, затінення та доступ до членів базового класу.

Тема 8. Поліморфізм: віртуальні методи, перевизначення (override) та пізні зв'язування об'єктів.

Тема 9. Абстракція: проектування за допомогою абстрактних класів та інтерфейсів, множинна реалізація інтерфейсів.

Тема 10. Узагальнення та колекції: використання Generics для створення типізованих структур даних та робота з бібліотекою системних колекцій.

Тема 11. Подійно-орієнтоване програмування: делегати, події та використання лямбда-виразів для обробки сигналів.

Тема 12. Надійність та архітектурні принципи: обробка виняткових ситуацій, основи Unit-тестування та вступ до принципів SOLID.

6. Система контролю та оцінювання.

Методи навчання:

- словесні методи (лекція, дискусія, пояснення);
- практичні (лабораторні роботи, написання коду, вирішення алгоритмічних задач);
- наочні методи (демонстрація, ілюстрація, презентація);
- робота з інформаційними ресурсами;
- самостійна робота.

Форми та методи оцінювання:

- усне опитування;
- письмове опитування;
- тестування;
- презентація результатів виконаних завдань;
- виконання практичних завдань;
- контрольні роботи;
- підсумковий контроль – екзамен.

Поточний контроль проводиться на кожному практичному занятті.

Об'єктом поточного контролю знань студентів є:

- систематичність, якість та своєчасність виконання і захисту лабораторних робіт;
- систематичність та своєчасність виконання завдань самостійної роботи студента;
- якість виконання модульних контрольних робіт.

Підсумковий контроль знань проводиться у формі екзамена.

На екзамен виносяться вузлові питання дисципліни та типові задачі, що потребують творчої відповіді та умінь синтезувати отримані знання і застосувати їх при вирішенні практичних задач. Максимально можлива оцінка за екзамен – 30 балів.

Семестрова кількість балів може становити від 0 до 100 балів і визначається як сума балів: отриманих за всі види роботи на лабораторних заняттях; за виконання самостійної роботи; модульних контрольних робіт, результат підсумкового контролю у формі екзамена.

Оцінювання здійснюється за національною шкалою – «відмінно», «добре», «задовільно», «незадовільно» та за шкалою ECTS.

Шкала оцінювання: національна та ЄКТС

Оцінка за шкалою ЄКТС	Оцінка за шкалою, що використовується у закладі вищої освіти та фахової передвищої освіти	Оцінка за національною шкалою
A	90-100	5 (відмінно)
B	80-89	4 (добре)
C	70-79	
D	60-69	3 (задовільно)
E	50-59	
FX	35-49	2 (незадовільно) з можливістю повторного складання
F	1-34	2 (незадовільно) з обов'язковим повторним вивченням дисципліни

Розподіл балів з навчальної дисципліни

Поточний контроль (аудиторна та самостійна робота)		Іспит	Загальна кількість балів
Модуль I	Модуль II		
35	35	30	100

Політика академічної доброчесності

Студент зобов'язаний ознайомитися з Положенням про забезпечення академічної доброчесності у ПВНЗ «Буковинський університет» та неухильно його дотримуватися. Текст документа розміщено у відкритому доступі на офіційному сайті університету. В освітньому процесі студент має виявляти дисциплінованість, ввічливість, доброзичливість, чесність і відповідальність.

Дотримання академічної доброчесності здобувачами освіти передбачає: самостійне виконання навчальних завдань, завдань підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей); посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; дотримання норм законодавства про авторське право і суміжні права; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації. Списування (копіювання тексту) під час виконання письмових робіт заборонені. Самостійні роботи у вигляді рефератів, доповідей, презентацій повинні мати коректні текстові посилання на використані інформаційні джерела. Дозволяється використання інструментів штучного інтелекту за умови дотримання принципів академічної доброчесності.

7.Рекомендована література

1. Олефіренко Н. В., Курганський А. Р., Остапенко Л. П., Пономарьова Н. О. Об'єктно-орієнтоване програмування мовою C# : навч. посіб. для здобувачів першого (бакалавр.) рівня вищ. пед. освіти / Н. В. Олефіренко, А. Р. Курганський, Л. П. Остапенко, Н. О. Пономарьова. – Харків : Харків. нац. пед. ун-т імені Г. С. Сковороди, 2024. – 253 с. (електрон. ресурс) URL: <https://dspace.hnpu.edu.ua/handle/123456789/14775> (дата звернення: 22.08.2025)

2. Нікуліна О. М., Іванов Л. В., Коцюба Н. В. Об'єктно-орієнтоване програмування мовою C# : метод. вказівки до лабораторних робіт з курсу «Об'єктно-орієнтоване програмування» : для студентів спец. 122 – Комп'ютерні науки, 126 – Інформаційні системи та технології / О. М. Нікуліна, Л. В. Іванов, Н. В. Коцюба. – Харків : Друкарня Мадрид, 2022. – 68 с. (електрон. ресурс) URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/57400> (дата звернення: 22.08.2025)

3. Настенко Д. В., Буханенко О. І., Марченко А. А. Об'єктно-орієнтоване програмування. Лабораторний практикум : навч. посіб. для студ. спец. 141 «Електроенергетика, електротехніка та електромеханіка» / Д. В. Настенко, О. І. Буханенко, А. А. Марченко. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 51 с. (електрон. ресурс) URL: <https://ela.kpi.ua/handle/123456789/48836> (дата звернення: 22.08.2025)

4. Price M. J. C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals. – 8th ed. – Packt Publishing, 2023. – 860 p.

5. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. – Apress, 2023 (Reprint). – 1600 p.

6. Albahari J. C# 12 in a Nutshell: The Definitive Reference. – O'Reilly Media, 2023. – 1080 p.

7. Nagel C. Professional C# and .NET. – 2024 Edition. – Wiley, 2024. – 1008 p.

8. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Prentice Hall, 2023 (Reprint).

9. Козуб Г.О., Осадчук С.І., Артеменко О.І. Паралельне програмування в мобільних іграх: методики оптимізації графіки та ігрової логіки Вчені записки Вернадського. Технічні науки Том 36 (75) № 4, 2025

10. Заяць В. М., Гаць Б. М., Осадчук С. І. Multi-agent approach to generation and analysis of computer graphics for pattern recognition tasks. Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова. 2025. № 4 (502). С. 280–288.

Інформаційні ресурси:

1. Microsoft Learn. .NET Documentation [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 22.08.2025).

2. GitHub. .NET Platform [Електронний ресурс]. – Режим доступу: <https://github.com/dotnet> (дата звернення: 22.08.2025).

3. Stack Overflow. Розділ “C#” [Електронний ресурс]. – Режим доступу: <https://stackoverflow.com/questions/tagged/c%23> (дата звернення: 22.08.2025).

4. Metanit. Посібник із C# [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/> (дата звернення: 22.08.2025).