



## СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

<b>Галузь знань</b>	12 Інформаційні технології
<b>Шифр та назва спеціальності</b>	121 Інженерія програмного забезпечення
<b>Назва освітньо-професійної програми</b>	Освітньо-професійна програма підготовки «Інженерія програмного забезпечення»
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Факультет</b>	Інформаційних технологій та кібербезпеки
<b>Кафедра</b>	Інженерії програмного забезпечення
<b>Статус навчальної дисципліни</b>	ОК-15 ОПП Освітньо-професійна програма підготовки «Інженерія програмного забезпечення»
<b>Форма навчання</b>	Денна

#### Викладачі

Косирева Ліаліна Анатоліївна  
[lialina@ukr.net](mailto:lialina@ukr.net)

Багачук Денис Геннадійович  
[bagachukdg@gmail.com](mailto:bagachukdg@gmail.com)



Старший викладач кафедри Інженерії програмного забезпечення

Доцент кафедри Інженерії програмного забезпечення,  
кандидат технічних наук

#### Загальна інформація про дисципліну

<b>Анотація до дисципліни</b>	Курс надає майбутнім фахівцям теоретичні знання та практичні навички з об'єктно-орієнтованого програмування та дозволяє сформувавши найважливіші практичні вміння з використанням актуальних підходів до розробки програмних засобів в різних галузях сучасного суспільства. Вивчається одна з найпоширеніших у світі мов об'єктно-орієнтованого програмування – мова C#.
<b>Мета дисципліни</b>	Метою викладання навчальної дисципліни є формування базових знань про сучасні технології об'єктно-орієнтованого програмування, практичне засвоєння основних понять і концепцій об'єктно-орієнтованого програмування.

<b>Компетентності, формуванню яких сприяє дисципліна</b>	СК-13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. СК-19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. СК-23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. СК-26. Здатність до алгоритмічного та логічного мислення.
<b>Результати навчання</b>	ПРН-1 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. ПРН-3 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення. ПРН-5 Знати і застосовувати відповідні математичні поняття, методи доменного, системного й об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення. ПРН-15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.
<b>Обсяг дисципліни</b>	Загальний обсяг дисципліни: 8 кредитів ЄКТС (240 годин). Для денної форми навчання: лекції – 30 годин, практичні заняття – 64 години, самостійна робота – 146 годин.
<b>Форма підсумкового контролю</b>	Екзамен
<b>Терміни викладання дисципліни</b>	Дисципліна викладається у 1-му та 2-му семестрах 2-го року навчання

## Програма дисципліни

<b>Тема 1.</b>	<b><i>Мова C#. Консольні прикладні проекти C#</i></b> Співвідношення між мовами програмування C, C++ і C#. Вікно консолі. Типи даних та оператори в C#. Тестування та налагодження програм мовою C#. Одновимірні і багатовимірні масиви. Масиви масивів. Цикли foreach у масивах. Функції у мові C#. Передача аргументів за значенням і за посиланням у функції. Рядки. Клас string та його методи.
<b>Тема 2.</b>	<b><i>Класи та об'єкти у мові C#</i></b> Класи, об'єкти та властивості (properties) у мові C#. Доступ до членів класу. Методи, їх параметри. Ініціалізація екземплярів класів. Специфікатори доступу. Конструктори та деструктори. Ключове слово this. Перевантаження операцій. Формат перевантаження для унарного та бінарного операторів.
<b>Тема 3.</b>	<b><i>Успадковування та інкапсуляція, поліморфізм</i></b> Ієрархії нащадків. Управління механізмом доступу до членів базового класу. Особливості використання конструкторів і деструкторів при реалізації механізму успадкування. Виклик затінених методів і базового конструктора.

Перекриття методів. Використання модифікатора типу `static`. Віртуальні методи класів. Поняття поліморфного коду. Використання вказівників на похідні типи як основа динамічного поліморфізму. Абстрактні класи. Використання ключового слова `sealed` для відвертання спадкоємства.

**Тема 4.** *Візуальне програмування*

Візуальний інтерфейс середовища Visual Studio .NET. Палітра компонент. Вікно властивостей. Створення віконного додатку. Властивості та методи форми. Події, пов'язані з формою. Основні компоненти.

Додатки з кількома формами, додатки MDI. Компоненти-контейнери. Створення і використання динамічних компонент. Підключення подієвих функцій до динамічних компонент. Події клавіатури.

**Тема 5.** *Введення і виведення даних у файли, списки*

Класи для введення і виведення даних. Класи `File` і `Directory`. Класи `FileInfo` і `DirectoryInfo`. Імена шляху і відносні шляхи. Класи `StreamWriter` і `StreamReader`. Об'єкти цих класів. Клас `FileStream`. Позиція усередині файлу. Читання і запис у файл. Робота з `BinaryWriter` і `BinaryReader`. Компоненти `dataGridView`.

Списки. Черга, стік. Клас `List`. Клас `Stack`. Клас універсальної колекції `Collection`. Клас `DateTime` для роботи з датами та часом.

**Тема 6.** *Побудова графічних зображень у C#*

Графічний інструментарій C#. Простір імен `System.Drawing`. Клас `Graphics`. Ієрархії класів. Побудова графічних зображень з використанням ієрархії класів. Приклад ієрархії класів для геометричних зображень. Візуалізація руху.

**Тема 7.** *Делегати. Події. Інтерфейси*

Статичні та нестатичні методи. Делегати. Додавання методів в делегат. Події. Додавання обробників подій. Обробка черги повідомлень

Оголошення та реалізація інтерфейсів. Використання інтерфейсних посилань. Інтерфейсні властивості. Перевантаження інтерфейсних методів. Успадковування інтерфейсів. Явна реалізація членів інтерфейсу. Вибір між інтерфейсом і абстрактним класом. Додаток на програмування подій.

**Тема 8.** *Виняткові ситуації та власні компоненти*

Виняткові ситуації і відповідні їм класи у C#. Оператори `try` та `finally`. Використання декількох `catch`-блоків. Ієрархія класів виняткових ситуацій.

Створення і використання бібліотек класів. Приклади побудови бібліотек. Клас `Component`. Побудова нащадків візуальних компонент. Додавання властивостей і обробників подій.

## Список рекомендованих джерел

1. В.В.Бублик – Об’єктно-орієнтоване програмування, К.: ІТкнига, 2015. – 624 с.
2. Косирева Л. А. Електронний конспект лекцій з дисципліни «Алгоритмізація та програмування» - Одеса: ОНУ, 2021.
3. Коноваленко І.В. Програмування мовою С# 7.0: навчальний посібник / Коноваленко І.В., Марущак П.О., Савків В.Б. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017. 300 с. URL : <http://elartu.tntu.edu.ua/handle/lib/22436>.
4. Об’єктно-орієнтоване програмування. Частина 1. Основи об’єктно-орієнтованого програмування на мові С#: Навчальний посібник. / Д.В. Настенко, А. Б. Нестерко. – К.: НТУУ «КПІ», 2016. – 76 с. URL:
5. Збірник задач з програмування / Глазунова Л.В. та ін. – Одеса: ОНАЗ, 2011.
6. Troelsen A., Pro C# and the .NET Framework – Апресс, 2012. - 1487 с.
7. [https://ela.kpi.ua/bitstream/123456789/16671/1/OOP\\_manual.pdf](https://ela.kpi.ua/bitstream/123456789/16671/1/OOP_manual.pdf)

## Інформація про консультації

**Щосереди** на протязі семестра з 13<sup>00</sup> до 14<sup>20</sup> год., конференція ZOOM ст. викладача Косиревої Л.А.

## Загальна схема оцінювання

Сума балів за всі види навчальної діяльності	Шкала ЄКТС	Оцінка за національною шкалою		Нарахування балів	Бали нараховуються таким чином:  <i>Оцінювання знань здобувачів вищої освіти здійснюється за 100-бальною шкалою і становить: за поточну успішність (участь у практичних заняттях, виконання практичних завдань та контрольних робіт) – до 60 балів, за результати екзамену – до 40 балів.</i>
		для іспиту	для заліку		
90-100	A	Відмінно	зараховано		
82-89	B	Добре			
74-81	C				
64-73	D				
60-63	E	Задовільно			
35-59	FX	Незадовільно з можливістю повторного складання	Не зараховано з можливістю повторного складання		

0-34	F	Незадовільно з обов'язковим повторним вивченням дисципліни	Не зараховано з обов'язковим повторним вивченням дисципліни		
------	---	--	---	--	--

### Політика опанування дисципліни

**Відвідування:** Присутність на лекціях практичних, лабораторних заняттях та контрольних заходах (екзамен/залік) є обов'язковою. Відсутність на заняттях може бути лише з поважних причин.

**Дотримання принципів академічної доброчесності:** Підготовка усіх завдань, письмових робіт і т. ін., що виконуються в межах дисципліни, здійснюється здобувачем вищої освіти самостійно, на засадах академічної доброчесності. Викладач має право для перевірки робіт застосовувати програму **Unicheck**.

**Умови зарахування пропущених занять:** Зарахування пропущених практичних та лабораторних занять проводяться під час консультацій.

**Інші умови:** Навчально-методичні матеріали дисципліни розміщені на платформі Moodle, за посиланням [.....](#)