

Назва курсу	CSB009: Об'єктно-орієнтоване програмування
Викладач	Антонов Ю.С., к. фіз.-мат.н., доцент
Профайл викладача	https://scholar.google.com/citations?user=0Nl4cRYAAAAJ&hl=en
Контактний телефон	(+38 044) 455-69-82
E-mail	AntonovYuS@krok.edu.ua y.s.antonov.edu@gmail.com
Консультації	Згідно з розкладом (online), MS Teams

1. Коротка анотація до курсу

Метою курсу є формування знань студента щодо проектування та розробки програмного забезпечення із застосуванням об'єктно-орієнтованої парадигми програмування.

У курсі розглядаються принципи об'єктно-орієнтованого програмування (інкапсуляція, успадкування, поліморфізм), необхідність оптимального балансу між наслідуванням та композицією, специфіка використання абстрактних класів та інтерфейсів, елементи об'єктно-орієнтованого аналізу та проектування, принципи DRY, KISS, SOLID та шаблони проектування. Для створення об'єктно-орієнтованих комп'ютерних програм в курсі використовується мова програмування C# та інтегроване середовище розробки Visual Studio Community (JetBrains Rider, Visual Studio Code).

2. Результати навчання

Програмні компетентності та результати навчання, компоненти яких формуються та розвиваються у курсі

Внесок дисципліни “Об'єктно-орієнтоване програмування” в формування наступних програмних результатів навчання:

Програмний результат навчання	Результат навчання з дисципліни
ПР9 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. (спільно з дисциплінами: CSB003 Алгоритмізація та програмування, CSB014 Технології створення програмного забезпечення, CSB025 Технології розподілених систем та паралельних обчислень)	Студент може демонструвати вміння здійснювати об'єктно-орієнтований аналіз предметної області та подальше об'єктно-орієнтоване проектування програмного забезпечення з використанням мови UML та об'єктно-орієнтованих мов програмування;
ПР14 Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем. (спільно з дисципліною CSB003 Алгоритмізація та програмування)	Студент може: - розробляти програмне забезпечення із застосуванням об'єктно-орієнтованої парадигми програмування на мові програмування C#; - самостійно опановувати нові мови програмування, що підтримують об'єктно-орієнтовану парадигму програмування.

Передумовою формування результатів навчання з дисципліни є володіння здобувачем на достатньому рівні наступних компетентностей:

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК1);
- здатність спілкуватися іноземною мовою (ЗК5);
- здатність оцінювати та забезпечувати якість виконуваних робіт (ЗК12).

Спеціальні компетентності:

- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління (СК8).

3. Обсяг курсу

Вид заняття	150 годин / 5 кредитів ECTS		
	форма навчання		
	денна	заочна	дистанційна
лекції	28		
практичні заняття	14		
лабораторні роботи	14		
самостійна робота	60		
консультації	4		
підсумковий контроль	30		

4. Технічне обладнання і програмне забезпечення

Технічне обладнання: комп'ютер/ноутбук.

Програмне забезпечення:

- Операційна система: Windows 10 / 8.1
або Linux Debian 10 / 11 (<https://www.debian.org/CD/http-ftp/#stable>),
або Ubuntu 20.04 LTS (<https://ubuntu.com/download/desktop>)
- .NET Core 3.1 або .NET 5.0 (<https://dotnet.microsoft.com/download>);
- IDE:
Visual Studio Community (<https://visualstudio.microsoft.com/ru/thank-you-downloading-visual-studio/?sku=Community&rel=16>)
або JetBrains Rider (<https://www.jetbrains.com/rider/>)
або Visual Studio Code (https://code.visualstudio.com/?wt.mc_id=vscom_downloads)

5. Політики курсу

Студенти мають дотримуватись кодексу академічної доброчесності:
https://www.krok.edu.ua/download/nakazi/2018-10-18_kodeks-akademichnoi-dobrochesnosti.pdf

6. Схема курсу

Тиждень / кіль-ть годин	Тема	Форма діяльності	Завдання / кіль-ть годин	Термін виконання
Тиждень #1 /	Основи ООП на C#.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
8 годин	Приклади роботи зі структурами. Конструктори та деструктори. Робота з класами та об'єктами.	практичне заняття	Розв'язання практичних завдань / 2	
	Історія появи об'єктно-орієнтованих мов програмування	самостійна робота	Огляд рекомендованих джерел інформації. / 4	
Тиждень #2 /	Інкапсуляція та приховування.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень

8 годин	Створення класів та об'єктів. Блоки public, static, protected sealed класи	лабораторна робота	Звіт з виконання лабораторної роботи / 2	
		самостійна робота	Огляд рекомендованих джерел інформації / 4	
Тиждень #3 /	Наслідування.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
8 годин	Наслідування та композиція.	практичне заняття	Розв'язання практичних завдань / 2	
	Переваги та недоліки наслідування.	самостійна робота	Огляд рекомендованих джерел інформації / 4	
Тиждень #4 /	Поліморфізм.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	Наслідування та поліморфізм	лабораторна робота	Звіт з виконання лабораторної роботи / 2	
	Геометричні фігури: наслідування та поліморфізм.	самостійна робота	Огляд рекомендованих джерел інформації / 8	
Тиждень #5 /	Інтерфейси та абстрактні класи.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
10 годин	Створення абстрактних класів. Створення інтерфейсів.	практичне заняття	Розв'язання практичних завдань / 2	
	Використання стандартних інтерфейсів: ICloneable, IEnumerable, IComparable, IComparable	самостійна робота	Огляд рекомендованих джерел інформації / 6	
Тиждень #6 /	Елементи об'єктно-орієнтованого аналізу та проектування.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
10 годин		лабораторна робота	Звіт з виконання лабораторної роботи / 2	
		самостійна робота	Огляд рекомендованих джерел інформації. Об'єктно-орієнтований аналіз предметної області у індивідуальних завданнях. / 8	
Тиждень #7 /	Елементи об'єктно-орієнтованого аналізу та проектування	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин		практичне заняття	Розв'язання практичних завдань / 2	
		самостійна робота	Огляд рекомендованих джерел інформації. Об'єктно-орієнтоване проектування для індивідуальних завданнях. / 8	
Тиждень #8 /	Елементи об'єктно-орієнтованого аналізу та проектування	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин		лабораторна робота	Звіт з виконання лабораторної роботи / 2	
		самостійна робота	Огляд рекомендованих джерел інформації. Об'єктно-орієнтоване проектування для індивідуальних завданнях. / 8	
Тиждень #9 /	Принципи DRY, KISS, SOLID.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
10 годин		практичне заняття	Розв'язання практичних завдань / 2	
		самостійна робота	Огляд рекомендованих джерел інформації. Аналіз	

			власного коду та прикладів, на предмет порушення принципів DRY, KISS або SOLID / 8	
Тиждень #10 /	Реалізація проекту «MathLib Project»	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	об'єктно-орієнтованого проектування.	лабораторна робота	Звіт з виконання лабораторної роботи / 2	
	об'єктно-орієнтованого проектування.	самостійна робота	Огляд рекомендованих джерел інформації / 8	
Тиждень #11 /	Реалізація проекту «Гра хрестики нолики»	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	Концепція MVC	практичне заняття	Розв'язання практичних завдань / 2	
	Недоліки концепції MVC	самостійна робота	Огляд рекомендованих джерел інформації / 2	
Тиждень #12 /	Стандартні колекції та Generic типи.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	List<T>. LinkedList<T>. Stack<T>. Dictionary<T, V>. Створення власних Generic класів.	лабораторна робота	Звіт з виконання лабораторної роботи / 2	
	ArrayList. Queue<T>. ObservableCollection. Обмеження універсальних типів. Наслідування узагальнених типів.	самостійна робота	Огляд рекомендованих джерел інформації / 8	
Тиждень #13 /	Паттерни проектування.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	Factory Method. Abstract Factory. Singleton. Adapter(Wrapper). Facade	практичне заняття	Розв'язання практичних завдань / 2	
	Builder. Prototype. Composite. Decorator. Flyweight. Proxy.	самостійна робота	Огляд рекомендованих джерел інформації / 8	
Тиждень #14 /	Паттерни проектування.	лекція	Аналіз прикладів з лекції, підготовка запитань. / 2	1 тиждень
12 годин	Command. Iterator. Visitor. Template Method. Strategy. State.	лабораторна робота	Звіт з виконання лабораторної роботи / 2	
	Chain of Responsibility. Memento. Observer.	самостійна робота	Огляд рекомендованих джерел інформації / 4	

7. Система оцінювання та вимоги

Загальна система оцінювання курсу	Оцінювання проводиться за 100-бальною накопичувальною шкалою. Бали накопичуються за наступним співвідношенням: - лабораторні роботи – 14% семестрової оцінки (7 робіт по 2 бали кожна); - індивідуальні роботи – 60% семестрової оцінки (5 робіт що оцінюються у 10, 10, 10, 12 та 18 балів); - іспит – 26% семестрової оцінки.
Практичні заняття / лабораторні роботи	Здача вирішених практичних завдань і звітів з індивідуальних та лабораторних робіт повинна здійснюватися відповідно до встановлених дедлайнів. Виконані завдання завантажуються студентом у відповідні модулі на платформі Moodle.
Умови допуску до підсумкового контролю	Виконання всіх індивідуальних та лабораторних робіт є обов'язковою умовою для допуску до складання іспиту

8. Рекомендована література до курсу

Базовий підручник

Troelsen A., Japikse Ph. Pro C# 9 with .NET 5 (10th edition). Apress. 2021. 1383 p

Допоміжна література

1. Кравець П.О. Об'єктно-орієнтоване програмування: навч. Посібник. Львів: Видавництво Львівської політехніки, 2012. 624 с.
2. Ian Griffiths. Programming C# 8.0. O'Reilly Media. 2020 706 p.
3. Vaskaran Sarcar. Getting Started with Advanced C#: Upgrade Your Programming Skills. Apress. 2020. 412 p.
4. Gamma, E., Helm R., Johnson R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software, 1st edn. Addison-Wesley Professional (1994).
5. Craig Berg. C# Programming For Beginners & Intermediates: C# Made Easy Step By Step With Hands on Projects. Independently published. 2020. 130p.
6. Joseph Albahari, Ben Albahari. C# 7.0 Pocket Reference. O'Reilly. 2017. 237 p.
7. Eugene Agafonov. Multithreading with C# Cookbook. Packt Publishing. 2016. 228p.
8. Об'єктно-орієнтоване програмування. Методичні рекомендації до виконання індивідуальних завдань / Ю. С. Антонов, Ю. В. Шамарін. Вінниця: ДонНУ імені Василя Стуса, 2018. – 50с.

Онлайн-ресурси

1. C# documentation - <https://docs.microsoft.com/en-us/dotnet/csharp/>
2. C# - :<https://metanit.com/sharp/tutorial/1.1.php>

До уваги студентів: усі навчально-методичні матеріали (презентації лекцій, тренувальні тести, тренувальні практичні завдання, завдання тощо) подані в Moodle за посиланням <https://dist1.krok.edu.ua/course/view.php?id=20>.