



Кафедра комп'ютерної інженерії та інформаційних технологій

## СИЛАБУС

<b>Базова інформація про дисципліну</b>	
<b>Назва дисципліни</b>	<b>SE004 Об'єктно-орієнтоване програмування / Object-Oriented Programming</b>
<b>Рівень вищої освіти / фахової передвищої освіти</b>	Фахова передвища
<b>Галузь знань</b>	12 «Інформаційні технології»
<b>Спеціальність</b>	121 «Інженерія програмного забезпечення»
<b>Освітня програма</b>	Інженерія програмного забезпечення
<b>Семестр</b>	IV семестр
<b>Факультет / відділення</b>	Інженерії програмного забезпечення
<b>Курс</b>	2 курс (9 кл), 1 курс (11 кл.)
<b>Анотація курсу</b>	<p>Метою курсу є формування знань студента щодо проєктування та розробки програмного забезпечення із застосуванням об'єктно-орієнтованої парадигми програмування.</p> <p>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 проєктування.</p> <p>Для створення об'єктно-орієнтованих комп'ютерних програм в курсі використовується мова програмування C# та інтегроване середовище розробки Visual Studio.</p>
<b>Сторінка курсу в MOODLE</b>	<a href="http://78.137.2.119:2929/course/view.php?id=283">http://78.137.2.119:2929/course/view.php?id=283</a>
<b>Мова викладання</b>	українська
<b>Лектор курсу</b>	<p>Немченко Вікторія Юріївна, викладач</p> <p>Канали комунікації:</p> <p>СДН «Moodle»: повідомлення в чаті</p> <p>E-mail: victoria.nvy@gmail.com</p>
<b>Місце дисципліни в освітній програмі</b>	
<b>Освітня програма</b>	<a href="http://csbc.edu.ua/documents/otdel/koop_pr.pdf">http://csbc.edu.ua/documents/otdel/koop_pr.pdf</a>

<p><b>Перелік загальних компетентностей (ЗК)</b></p>	<p>Знання та розуміння предметної області та розуміння професійної діяльності.  Здатність до пошуку, оброблення та аналізу інформації з різних джерел.  Здатність до абстрактного мислення, аналізу та синтезу.  Здатність застосовувати знання у практичних ситуаціях.  Здатність спілкуватися іноземною мовою;  Здатність оцінювати та забезпечувати якість виконуваних робіт.  Здатність вчитися і оволодівати сучасними знаннями;  Здатність до пошуку, оброблення та аналізу інформації з різних джерел;  Здатність працювати в команді;</p>
<p><b>Перелік спеціальних компетентностей (СК)</b></p>	<p>Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, 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.  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.  Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;  Здатність до алгоритмічного та логічного мислення.</p>
<p><b>Перелік програмних результатів навчання</b></p>	<p>Вміти застосовувати методи обчислення та структури даних для вирішення задач аналізу та синтезу алгоритмів.  Вміти систематизувати та узагальнювати інформація про підходи, методи та засоби розробки супроводу програмного забезпечення, вдосконалювати відповідні знання, вміння і навички протягом усього життя.</p>

	<p>Вміти знаходити аналогії та застосовувати знання, вміння та навички з суміжних дисциплін для формування та вирішення професійних завдань.</p> <p>Знати і застосовувати методи доменного, системного і об'єктно-орієнтованого аналізу для розробки програмного забезпечення.</p> <p>Знати основні стандарти, принципи та нотації проектування програмного забезпечення та вміти застосовувати їх на практиці для моделювання структури, поведінки та внутрішніх процесів програмної системи.</p> <p>Вміти здійснювати процес впровадження та супроводу програмних систем.</p>
<b>Опис дисципліни</b>	
<b>Структура навантаження на студента</b>	<p>Загальна кількість годин – 90</p> <p>Кількість кредитів – 3</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p> <p>Кількість годин для самостійної роботи студентів – 30</p> <p>Форма підсумкового контролю – залік</p>
<b>Методи навчання</b>	<p>Словесні (інформаційна, самостійна робота з джерелами інформації, науково-популярна розповідь);</p> <p>Наочні (презентаційні повідомлення)</p> <p>Практичні (лабораторні роботи);</p> <p>Інтерактивні методи (дистанційні консультації).</p>
<b>Зміст дисципліни</b>	
Тема 1. Введення в об'єктно-орієнтоване програмування.	Історія розвитку ООП. Об'єктно-орієнтоване проектування.
Тема 2. Основи об'єктно-орієнтованого програмування.	Поняття класів та об'єктів. Визначення класу. Область дії класу та доступ до елементів класу. Функції доступу та сервісні функції. Робота з конструкторами та деструкторами. Статичні методи та поля. Використання елементів-даних та елементів-функцій.
Тема 3. Основні принципи реалізації класів.	Константні об'єкти та константні елементи-функції. Композиція класів. Дружні функції та дружні класи. Операції виділення пам'яті new, delete. Статичні елементи класу. Абстракція даних та укриття інформації. Контейнерні класи та ітератори.

Тема 4. Введення в поліморфізм	Перевантаження операцій. Використання дружніх структур. Основні принципи перевантаження операцій. Заборони на перевантаження операцій. Функції-операції як елементи класу і як дружні функції. Перевантаження одномісних операцій. Перевантаження двомісних операцій. Перетворення типів.
Тема 5. Наслідування (спадкування).	Базові та похідні класи. Захищені елементи. Перетворення покажчиків базового класу у покажчики на похідний клас. Перевизначення елементів базового класу у похідному класі. Відкриті, захищені та закриті базові класи. Безпосередні та непрямі базові класи. Використання конструктора і деструктора у похідному класі. Неявне перетворення об'єктів похідного класу до базового. Спадкування у конструюванні програмного забезпечення. Складне спадкування.
Тема 6. Віртуальні функції і поліморфізм	Віртуальні функції. Абстрактні базові класи і реальні класи. Поліморфізм. Нові класи і динамічне зв'язування. Віртуальні деструктори.
Тема 7. Обробка виняткових ситуацій	Основні поняття та використання виняткових ситуацій. Стек викликів, стек блоків. Обробка виняткових ситуацій стандартними засобами. Обробка виняткових ситуацій кастомізованими засобами.
Тема 8. Об'єктно-орієнтована технологія програмування на мові C#	Базові поняття мови C#. Основи синтаксису C#, система типів .NET. Масиви, синтаксис оголошення, ініціалізація, застосування масивів. Класи, інтерфейси, наслідування, поліморфізм Об'єкти, створення об'єкта. Конструктор, деструктор. Основи перевантаження операторів. Наслідування і поліморфізм. Інтерфейси.
<b>Політика дисципліни</b>	
<b>Політика відвідування</b>	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання організується в он-лайн формі за погодженням із керівником курсу.

<b>Політика щодо дедлайнів та перескладання</b>	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
<b>Академічна доброчесність</b>	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.

### **Система оцінювання**

Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку, відповідно до графіку навчального процесу. Підсумкова оцінка за умови заліку виставляється як загальна сума балів, набраних за результатами поточного контролю.

### **Накопичування рейтингових балів з навчальної дисципліни 4-й семестр (9 кл), 2-й семестр (11 кл)**

<b>Види навчальної роботи</b>	<b>Мах кількість балів</b>
<b>Аудиторна</b>	
Практичні завдання (10 тем)	80
Тестування	20
<b>Разом</b>	<b>100</b>

### **Шкала оцінювання**

<b>ECTS</b>	<b>Бали</b>	<b>Зміст</b>
<b>A</b>	90-100	Бездоганна підготовка в широкому контексті
<b>B</b>	80-89	Повні знання, міцні вміння
<b>C</b>	70-79	Хороші знання та вміння
<b>D</b>	65-69	Задовільні знання, стереотипні вміння
<b>E</b>	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
<b>FX</b>	35-59	Слабкі знання, відсутність умінь
<b>F</b>	1-34	Необхідний повторний курс

**Список рекомендованих джерел**

1. Коноваленко І.В. Програмування мовою С# 7.0 : навчальний посібник / Коноваленко І.В., – Тернопіль:Тернопільський національний технічний університет імені Івана Пулюя 2017. 300 с.
2. Щербаков О.В. Основи об'єктно-орієнтованого програмування : навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2019. 246 с.
3. Troelsen A., Japikse Ph. Pro C# 9 with .NET 5 (10th edition). Apress. 2021. 1383 p
4. Ian Griffiths. Programming C# 8.0. O'Reilly Media. 2020. 706 p.
5. Vaskaran Sarcar. Getting Started with Advanced C#: Upgrade Your Programming Skills. Apress. 2020. 412 p.
6. Gamma, E., Helm R., Johnson R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software, 1st edn. Addison-Wesley Professional (1994).
7. Craig Berg. C# Programming For Beginners & Intermediates: C# Made Easy Step By Step With Hands on Projects. Independently published. 2020. 130p.
8. Joseph Albahari, Ben Albahari. C# 7.0 Pocket Reference. O'Reilly. 2017. 237p.
9. Eugene Agafonov. Multithreading with C# Cookbook. Packt Publishing. 2016. 228p.
10. <https://learn.microsoft.com/en-us/dotnet/csharp/>
11. C# - :<https://metanit.com/sharp/tutorial/1.1.php>
12. Stroustrup B. The C++ Programming Language / B. Stroustrup. Addison-Wesley Professional, 2013, 1376 p.