

**БІЛГОРОД-ДНІСТРОВСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ПРИРОДОКОРИСТУВАННЯ, БУДІВНИЦТВА ТА
КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ**

Циклова комісія інформаційних технологій



ЗАТВЕРДЖЕНО

Заступник директора з

навчальної роботи

 **Марина ЗАЙЧЕНКО**

«29» серпня 2024 р.

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

здобувачів освіти спеціальності

121 Інженерія програмного забезпечення

Білгород-Дністровський, 2024

Програма навчальної дисципліни «**Об'єктно-орієнтоване програмування**» складена відповідно до освітньо-професійної програми для здобувачів освіти зі спеціальності **121 Інженерія програмного забезпечення**.

Розробник: Боцан Марія Володимирівна, викладач інформаційних технологій, кваліфікаційна категорія «спеціаліст I категорії»

Робоча програма розглянута та схвалена на засіданні циклової комісії інформаційних технологій

Протокол № 1 від 29.08.2024 року

Голова циклової комісії _____ /Сергій ТІТЯПКИН/

Схвалено методичною радою Білгород-Дністровського фахового коледжу природокористування, будівництва та комп'ютерних технологій

Протокол № __6__ від 29.08.2024 року

Голова методичної ради _____ /Марина ЗАЙЧЕНКО/

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Компонент освітньої програми, спеціальність, освітньо-професійний ступінь	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
<p>Кількість кредитів: 7,5 (окремо на курсовий проєкт: 2,0)</p> <p>Модулів: 3</p> <p>Змістових модулів: 4</p> <p>Загальна кількість годин: 225 (окремо на курсовий проєкт: 60)</p>	<p>Компонент освітньої програми</p> <p><i>Цикл професійної підготовки</i></p> <p>Спеціальність</p> <p><i>121 Інженерія програмного забезпечення</i></p> <p>Освітньо-професійний ступінь</p> <p><i>Фаховий молодший бакалавр</i></p>	Вибіркова	
		Рік підготовки:	
		3й, 4й	-
		Семестр	
		5й, 6й, 7й	-
		Лекційні заняття:	
		47 год.	-
		Лабораторні заняття	
		90 год.	-
		Самостійна робота	
		88 год.	-
		Курсовий проєкт	
		60 год.	-
		Індивідуальні заняття:	
		-	-
Вид контролю:			
<p><u>диференційований залік</u> (5й семестр), <u>диференційований залік</u> (6й семестр), <u>екзамен</u> (7й семестр), <u>захист курсового проєкту</u> (7й семестр) для денної форми</p>			
-			

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна передбачена структурно-логічною схемою підготовки фахівців освітньо-професійного ступеню «Фаховий молодший бакалавр» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Метою викладання навчальної дисципліни «Об'єктно-орієнтоване програмування» є формування у здобувачів освіти ґрунтовних знань з об'єктно-орієнтованої парадигми програмування, що є однією з найпоширеніших концепцій розробки програмного забезпечення, а також оволодіння методами створення сучасних програмних продуктів із використанням принципів ООП. Вивчення дисципліни сприяє розумінню студентами принципів проєктування програмного забезпечення, розробки архітектури застосунків, підходів до роботи з класами, об'єктами, інтерфейсами та бібліотеками.

Завдання вивчення дисципліни полягає у формуванні компетентностей, необхідних для ефективного використання концепцій ООП у розробці програмного забезпечення. Студенти набувають знань про ключові принципи об'єктно-орієнтованого програмування, а також отримують практичні навички їх застосування. Особливу увагу приділено роботі з мовою програмування C# та платформою .NET.

Предметом вивчення дисципліни є принципи побудови об'єктно-орієнтованих програм, методи розробки програмного коду, організація програмного середовища C#, створення віконних застосунків, робота з подіями, файлами та базами даних. Також розглядаються питання ефективного управління пам'яттю, паралельного виконання задач, мережевої взаємодії та використання сучасних підходів до розробки програмного забезпечення.

Міждисциплінарні зв'язки: «Основи програмної інженерії», «Алгоритми і структури даних», «Основи програмування», «Організація баз даних», «Операційні системи», «Програмування для мобільних пристроїв», «Навчальна практика з об'єктно-орієнтованого програмування».

Набуті здобувачами освіти компетенції згідно з вимогами освітньо-професійної програми «Інженерія програмного забезпечення»:

Загальні

ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК7. Здатність застосовувати знання у практичних ситуаціях.

Спеціальні

СК1. Здатність алгоритмічно та логічно мислити.

СК3. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.

СК5. Здатність брати участь у визначенні та формулюванні вимог до програмного забезпечення.

СК6. Здатність брати участь у проєктуванні програмного забезпечення.

СК7. Здатність розробляти модулі і компоненти програмного забезпечення

за допомогою типових алгоритмів та інструментів.

СК9. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.

Структура навчальної дисципліни є орієнтовною. Під час складання навчальних програм викладачі навчальних закладів можуть вносити обґрунтовані зміни та доповнення в зміст програмного матеріалу і розподіл навчальних годин за темами в межах бюджетного часу, відведеному навчальним планом на вивчення дисципліни. Внесені зміни повинні бути обговорені на засіданні циклової комісії і затверджені заступником директора з навчальної роботи.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Модуль 1. Основи об'єктно-орієнтованого підходу до розробки програмного забезпечення.

Змістовий модуль 1. Особливості об'єктно-орієнтованого програмування (ООП) та мова C#.

Тема 1.1. Об'єктно-орієнтоване мислення при програмуванні

Ознайомлення з концепціями об'єктно-орієнтованого мислення, що лежать в основі сучасного програмування. Розглядаються принципи моделювання реального світу через об'єкти, їх властивості та взаємодію для створення гнучких і масштабованих програмних систем.

Тема 1.2. Мова програмування C# та платформа .NET

Огляд мови програмування C#, її основних конструкцій та особливостей. Архітектура платформи .NET, принципи роботи CLR (Common Language Runtime). Використання середовища розробки для створення додатків.

Лабораторна робота: Створення та компіляція першої консольної програми на C#.

Тема 1.3. Створення та використання класів і об'єктів

Поняття класу та об'єкта в ООП. Оголошення класів у C#, створення екземплярів, конструктори та деструктори. Використання полів, методів та властивостей.

Лабораторна робота: Розробка класу та створення об'єктів у C#.

Тема 1.4. Реалізація трьох основних принципів ООП: інкапсуляція, наслідування та поліморфізм

Детальний розгляд інкапсуляції, наслідування та поліморфізму. Доступ до членів класу.

Абстрактні класи та інтерфейси в C#.

Лабораторна робота: Реалізація інкапсуляції на прикладах.

Лабораторна робота: Реалізація успадкування на прикладах.

Лабораторна робота: Реалізація поліморфізму на прикладах.

Змістовий модуль 2. Розробка подійно-орієнтованих додатків за допомогою технології Windows Forms

Тема 2.1. Основи технології Windows Forms та створення віконних додатків

Огляд Windows Forms. Створення графічного інтерфейсу користувача (GUI). Використання стандартних елементів керування (кнопки, текстові поля, мітки тощо).

Лабораторна робота: Розробка простого віконного додатка на Windows Forms.

Тема 2.2. Обробка подій та інтерактивність у Windows Forms

Обробка подій у Windows Forms. Використання обробників подій для керування взаємодією з користувачем.

Лабораторна робота: Додавання інтерактивності до віконного додатка.

Лабораторна робота: Робота з групами елементів керування: RadioButton, CheckBox, GroupBox.

Лабораторна робота: Використання вкладок та панелей (TabControl, Panel).

Лабораторна робота: Робота зі списками та таблицями: ListBox, ComboBox, DataGridView.

Тема 2.3. Обробка винятків та управління помилками

Обробка винятків у C#. Використання try-catch-finally. Генерація власних винятків.

Лабораторна робота: Реалізація механізму обробки винятків у програмі.

Модуль 2. Розробка багатофункціональних програмних додатків у C#.NET.

Змістовий модуль 3. Збереження, обробка та управління даними у C#.NET

Тема 3.1. Колекції та робота зі структурами даних у C#

Використання списків, словників, черг, стеків у C#. Різниця між різними структурами даних, особливості використання.

Тема 3.2. Робота з файлами та потоками в C#

Операції з файлами у C#. Потокове введення/виведення. Читання та запис текстових і бінарних файлів.

Лабораторна робота: Робота з файлами та потоками у C#.

Тема 3.3. Серіалізація та десеріалізація даних у C#

Основи серіалізації та десеріалізації об'єктів у C#. Використання форматів JSON, XML та Binary для збереження та передачі даних. Огляд механізмів серіалізації за допомогою бібліотек .NET, включаючи System.Text.Json та System.Xml.Serialization. Практичне застосування серіалізації для збереження стану об'єктів і взаємодії між компонентами програмного забезпечення.

Лабораторна робота: Реалізація серіалізації та десеріалізації об'єктів у бінарному форматі.

Лабораторна робота: Реалізація серіалізації та десеріалізації об'єктів у форматах JSON та XML.

Тема 3.4. Використання бази даних SQLite при програмуванні C#

Основи роботи з базами даних SQLite у C#. Виконання SQL-запитів у додатках. Використання ADO.NET для підключення до бази даних.

Лабораторна робота: Взаємодія з базою даних SQLite через програмний додаток.

Тема 3.5. Принципи багатопотоковості та асинхронне програмування

Основи роботи з потоками у C#. Використання Thread, Task та async/await для створення багатопотокових і асинхронних додатків. Управління конкурентним доступом до ресурсів, синхронізація потоків, запобігання стану гонки та використання lock, Mutex, Semaphore. Аналіз переваг та викликів асинхронного програмування для підвищення продуктивності застосунків.

Лабораторна робота: Реалізація багатопотокової програми з використанням потоків та асинхронних операцій у C#.

Тема 3.6. Робота з мережевими додатками та протоколами

Основи мережевого програмування у C#. Використання TCP та UDP для обміну даними між клієнтом і сервером. Застосування HttpClient для виконання HTTP-запитів та взаємодії з веб-сервісами. Розгляд принципів побудови клієнт-серверних застосунків та безпечної передачі даних у мережі.

Лабораторна робота: Створення клієнт-серверних додатків на основі UDP у C#.

Лабораторна робота: Створення клієнт-серверних додатків на основі TCP у C#.

Тема 3.7. Використання зовнішніх API для інтеграції з іншими сервісами

Підключення до сторонніх сервісів через API у C#. Використання REST API та протоколу HTTP для взаємодії з веб-службами. Методи автентифікації, передача параметрів, обробка відповідей у форматах JSON та XML. Огляд популярних бібліотек для роботи з API, зокрема HttpClient та RestSharp.

Лабораторна робота: Отримання та обробка даних із зовнішнього API, створення клієнта для інтеграції з веб-службою.

Модуль 3. Шаблони для об'єктно-орієнтованого проєктування.

Змістовий модуль 4. Архітектурні підходи та шаблони проєктування в C#

Тема 4.1. Поняття про шаблони проєктування і їх застосування.

Породжувальні шаблони.

Основи шаблонів(патернів) проєктування, їх класифікація та призначення. Огляд основних категорій: породжувальні, структурні та поведінкові шаблони. Важливість шаблонів у розробці масштабованих, підтримуваних та ефективних програмних рішень за об'єктно-орієнтованою методологією.

Шаблони створення об'єктів: Singleton, Factory Method, Abstract Factory, Builder, Prototype. Вибір відповідного шаблону залежно від задачі. Реалізація та приклади використання в реальних проєктах.

Лабораторна робота: Аналіз структури та принципів роботи шаблонів проєктування. Реалізація базового прикладу одного з найпростіших шаблонів Singleton у C#

Лабораторна робота: Реалізація породжувальних шаблонів у C#.

Тема 4.2. Структурні шаблони проєктування

Організація взаємодії між класами та об'єктами за допомогою структурних шаблонів: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy. Особливості застосування та переваги використання у великих системах.

Лабораторна робота: Реалізація структурних шаблонів у C#.

Тема 4.3. Поведінкові шаблони проєктування

Моделювання взаємодії між об'єктами та управління потоками виконання: Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor. Вибір оптимального шаблону для конкретної задачі.

Лабораторна робота: Реалізація поведінкових шаблонів у C#.

КУРСОВИЙ ПРОЄКТ

Мета виконання курсового проєкту - закріплення та поглиблення теоретичних знань, набуття студентами навичок практичної роботи з аналізу вимог, проектування та розробки програмного забезпечення.

Тематику курсового проєкту визначає циклова комісія навчального закладу згідно з рекомендаціями методичних об'єднань. Кожна тема проєкту має відповідати сучасним завданням з автоматизації та програмування в галузі комп'ютерної інженерії.

Виходячи з цього, завдання курсового проектування полягає в тому, щоб сформувані у студентів уміння застосовувати об'єктно-орієнтовані підходи до розробки програмного забезпечення, розробляти архітектуру програмних систем, працювати з інструментами моделювання, застосовувати принципи проектування та шаблони програмування.

Крім того, курсовий проєкт має сприяти розвитку навичок роботи з сучасними середовищами розробки, бібліотеками та фреймворками, налагодження та тестування програмного коду, а також оформлення технічної документації відповідно до встановлених стандартів.

Результатом виконання курсового проєкту має бути створення функціонального програмного продукту, що відповідає поставленим вимогам, а також здобуття практичного досвіду роботи над повним циклом розробки програмного забезпечення., підрахувати специфікацію арматури та відомість витрати сталі на елемент.

Склад і оформлення курсового проєкту

Курсовий проєкт складається з розробленого програмного продукту та пояснювальної записки, оформленої відповідно до методичних вказівок. Пояснювальна записка містить опис постановки задачі, аналіз предметної області, обґрунтування вибору методів та засобів розробки, опис архітектури та реалізації проєкту, результати тестування, а також висновки та список використаних джерел.

Графічна частина

Графічна частина проєкту розробляється у вигляді схем, діаграм та ілюстративних матеріалів, що відображають структуру програмного забезпечення, взаємодію його компонентів та алгоритмічні рішення. До графічної частини можуть входити:

- діаграми класів, послідовностей та компонентів (UML);
- блок-схеми алгоритмів;
- схеми бази даних;
- графічний інтерфейс користувача (якщо передбачено проєктом);
- результати тестування у вигляді таблиць, графіків або логів виконання.

Текстова частина

Текстова частина пояснювальної записки містить структурований опис розробленого програмного продукту, що включає:

- постановку задачі;
- аналіз аналогів та вибір технологій;

- проектування архітектури та основних модулів;
- опис реалізації та коду;
- тестування та результати перевірки працездатності;
- висновки та перспективи подальшого розвитку проєкту;
- літературні джерела;
- додатки(лістинг програмного коду).

Оформлення записки виконується згідно зі встановленими вимогами закладу освіти.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Об'єктно-орієнтоване програмування»

Назви розділів і тем	Кількість годин											
	Денна форма						Заочна форма					
	загальний обсяг	аудиторні					самостійна робота	загальний обсяг	аудиторні			самостійна робота
		в с ь о г о	з них			теоретичні			практичні	лабораторні	в с ь о г о	
теоретичні	практичні		лабораторні	теоретичні	практичні		лабораторні					
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1. Основи об'єктно-орієнтованого підходу до розробки програмного забезпечення.												
Змістовий модуль 1. Особливості об'єктно-орієнтованого програмування (ООП) та мова C#												
1.1 Об'єктно-орієнтоване мислення при програмуванні	4	2	2	0	0	2						
1.2 Мова програмування C# та платформа .NET	6	4	2	0	2	2						
1.3 Створення та використання класів і об'єктів	12	10	4	0	6	2						
1.4 Реалізація трьох основних принципів ООП: інкапсуляція, наслідування та поліморфізм	20	18	6	0	12	2						
Разом за змістовим модулем 1	42	34	14	0	20	8						
Змістовий модуль 2. Розробка подійно-орієнтованих додатків за допомогою технології Windows Forms												
2.1 Основи технології Windows Forms та створення віконних додатків	10	8	2	0	6	2						
2.2 Обробка подій та інтерактивність у Windows Forms	18	16	4	0	12	2						
2.3 Обробка винятків та управління помилками	10	6	2	0	4	4						
Разом за змістовим модулем 2	38	30	8	0	22	8						
Разом за модулем 1	80	64	22	0	42	16						
Модуль 2. Розробка багатофункціональних програмних додатків у C#.NET												
Змістовий модуль 3. Збереження, обробка та управління даними у C#.NET												
3.1 Колекції та робота зі структурами даних у C#	12	6	2	0	4	6						
3.2 Робота з файлами та потоками в C#	12	6	2	0	4	6						
3.4 Використання бази даних SQLite при програмуванні C#	14	6	2	0	4	8						

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>	<i>13</i>
3.5 Принципи багатопотоковості та асинхронне програмування	16	8	4	0	4	8						
3.6 Робота з мережевими додатками та протоколами	18	10	4	0	6	8						
3.7 Використання зовнішніх API для інтеграції з іншими сервісами	18	9	4	0	5	9						
<i>Разом за змістовим модулем 3</i>	90	45	18	0	27	45						
<i>Разом за модулем 2</i>	90	45	18	0	27	45						
Модуль 3. Шаблони для об'єктно-орієнтованого проєктування												
Змістовий модуль 4. Архітектурні підходи та шаблони проєктування в C#												
4.1 Поняття про шаблони проєктування і їх застосування. Породжувальні шаблони.	19	12	3	0	9	7						
4.2 Структурні шаблони проєктування	18	8	2	0	6	10						
4.3 Поведінкові шаблони проєктування	18	8	2	0	6	10						
<i>Разом за змістовим модулем 4</i>	55	28	7	0	21	27						
<i>Разом за модулем 3</i>	55	28	7	0	21	27						
ВСЬОГО З ДИСЦИПЛІНИ	225	137	47	0	90	88						
<i>Курсовий проєкт з ООП (окремо)</i>	60	-	-	-	-	60						

5. МЕТОДИ ТА ФОРМИ НАВЧАННЯ

I. Методи організації та здійснення навчально-пізнавальної діяльності

Словесні методи (бесіда, розповідь, пояснення, лекції тощо) характерні тим, що інформацію для засвоєння здобувач освіти отримує вербальними засобами, тобто через слово.

Наочні методи - інформація для засвоєння одержується на основі сенсорно-перцептивної діяльності (демонстрування, ілюстрації, показ об'єкта, віртуальної моделі).

Практичні методи. Суть їх у тому, що шляхом виконання практичних дій здобувач освіти отримує деяку інформацію, яку аналізує, робить висновок і приходить до тих знань, які необхідно засвоїти. Особливість методу в тому, що діяльність з одержання знань накладається в часі на діяльність з їх застосування, що дає винятково важливий педагогічний ефект.

II. Методи стимулювання інтересу до навчання і мотивації навчально-пізнавальної діяльності.

1. Бесіда, або діалог з аудиторією. Ставиться серія запитань, які потребують відповіді. Це дає можливість зрозуміти, чи готові здобувачі освіти сприймати новий матеріал, чи їх потрібно активізувати. Практика підказує, що здобувачі освіти ідуть на заняття не підготовлені, але коли знають, що буде опитування – готуються. Разом з тим це дає можливість виявити прогалини, що важливо не стільки для здобувача освіти, як для викладача.

2. Сократична бесіда. Ставиться серія запитань, які дають можливість здобувачу освіти дати не повну відповідь, що спонукає з зацікавленістю сприймати новий матеріал.

3. Проблемне заняття. Висловлюється проблема, з метою викликати зацікавленість у здобувачів освіти. Цей вид інтерактивних технологій можна використовувати після опрацювання серії занять, бо здобувачі освіти вже повинні мати багаж знань.

4. Дискусія. Відбувається активний обмін думками. Це різновид проблемних лекцій. Проводяться ділові ігри, самостійна робота. Лекція-дискусія дає можливість охопити складний, великий за обсягом і найбільш вдалий матеріал.

5. Аналіз конкретних ситуацій. Береться конкретна ситуація з життя (професійна діяльність, соціум тощо) і вирішується різними шляхами. Сьогодні неможливо навчати здобувача освіти старими методами. Знань стало так багато, професійні навички стали настільки багатоманітними, що їх неможливо передати в повному обсязі в межах традиційних методів, шляхом ретрансляції, позбавленої емоційності.

6. Заняття з використанням техніки зворотного зв'язку. Після подачі лекції починається її обговорення. З'ясовується наскільки здобувачі освіти зрозуміли матеріал.

7. Метод «заверши фразу». Здобувач освіти може продовжувати її своїми словами, а не так як у конспекті.

8. Консультація. Для індивідуальної роботи зі здобувачами освіти використовують **пояснення**.

6. МЕТОДИ ТА ФОРМИ КОНТРОЛЮ

За місцем у навчальному процесі розрізняють **вхідний, поточний, періодичний, підсумковий види контролю.**

Вхідний контроль – використовують перед вивченням нової теми на початку семестру для з'ясування загального рівня підготовки здобувачів освіти з дисципліни, щоб передбачити організацію їх навчально-пізнавальної діяльності.

Поточний контроль – спостереження викладача за навчальною діяльністю здобувачів освіти на занятті. Метою його є отримання оперативних даних про рівень знань здобувачів освіти і якість навчальної роботи на занятті, оптимізація управління навчальним процесом.

Періодичний (тематичний) контроль – виявлення й оцінювання засвоєних на кількох попередніх заняттях знань, умінь здобувачів освіти з метою визначення, наскільки успішно вони володіють системою знань, чи відповідають ці знання програмі. Різновидом періодичного є **тематичний контроль**, що полягає у перевірці та оцінюванні знань здобувачів освіти з кожної теми і спрямований на те, щоб усі належно засвоїли кожну тему.

Підсумковий контроль здійснюється наприкінці семестру або навчального року. Підсумкову оцінку за семестр виставляють за результатами тематичного оцінювання, за рік – на основі семестрових оцінок.

Навчальні досягнення здобувачів освіти з навчальної дисципліни «Об'єктно-орієнтоване програмування» можуть оцінюватися за кредитно-трансферною системою ЄКТС, в основу якої покладено принцип прозорості, об'єктивності, індивідуальності та певної уніфікованості. Головне завдання при цьому – досягти найбільш ефективного та об'єктивного оцінювання, яке повинне одночасно виконувати контролюючу й мотивуючу функції.

Кожен модуль включає лекційні та лабораторні заняття, самостійну роботу.

Модульний контроль знань здобувачів освіти здійснюється через проведення аудиторних письмових контрольних робіт та комп'ютерного тестування.

Кількість балів за роботу з теоретичним матеріалом, на практичних заняттях, під час виконання самостійної роботи залежить від дотримання таких вимог:

- своєчасність виконання завдань;
- повний обсяг їх виконання;
- якість виконання навчальних завдань;
- самостійність виконання;
- творчий підхід до виконання завдань;
- ініціативність у навчальній діяльності.

Форма підсумкового контролю успішності навчання – диференційований залік – 5-й семестр та бй семестр 3го року навчання, екзамен - 7-й семестр 4го року навчання (денна форма).

7. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання навчальних досягнень здобувачів освіти здійснюється відповідно до «Положення про оцінювання навчальних досягнень здобувачів освіти у Білгород-Дністровському фаховому коледжі природокористування, будівництва та комп'ютерних технологій».

Формами семестрової атестації є диференційований залік – 5й та 6й семестри 3го року навчання та екзамен - 7й семестр 4го року навчання (денна форма), а також окремо захист курсового проєкту – 7й семестр 4го року навчання (денна форма).

Результати навчання здобувачів фахової передвищої освіти Коледжу з теоретичної та практичної підготовки можуть оцінюватись за 100-бальною шкалою, оцінкою в ЄКТС.

Відповідно рейтинг здобувача освіти із засвоєння навчальної дисципліни може складатися з рейтингу з навчальної роботи – 70 балів та рейтингу з атестації – 30 балів. Таким чином, на оцінювання засвоєння змістових модулів, на які поділяється навчальний матеріал дисципліни, передбачається 70 балів. Рейтингові оцінки із змістових модулів, як і рейтинг з атестації, теж обчислюються за 100-бальною шкалою.

Для занесення оцінок до екзаменаційної відомості, індивідуального навчального плану здобувача освіти (залікової книжки) та журналу рейтингової оцінки знань здобувача освіти його рейтинг з різних видів навчальної роботи у балах переводиться у національну та ЄКТС (Європейська кредитна трансферно-накопичувальна система) оцінки згідно з таблицею.

Відповідність результатів контролю знань за різними шкалами і критерії оцінювання

Оцінка ЄКТС	Сума балів за 100 бальною шкалою	Національна шкала (12-бальна)	Національна шкала (4-бальна)	Рівень компетентності	Критерії оцінювання
A	90 – 100 (відмінно)	12-10	відмінно	Високий рівень	Здобувач освіти виявляє особливі творчі здібності, вміє самостійно здобувати знання, без допомоги викладача знаходить та опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для ухвалення рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні обдарування і нахили.
B	85 – 89 (дуже добре)	9-8	добре	Достатній рівень	Здобувач освіти вільно володіє вивченим обсягом матеріалу, застосовує його на практиці, вільно розв'язує справи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна.
C	75 – 84 (добре)	7			Здобувач освіти вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача; в цілому самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, серед яких є суттєві, добирати аргументи для підтвердження думок.
D	70 – 74 (задовільно)	6-5	задовільно	Середній рівень	Здобувач освіти відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень; з допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, серед яких є значна кількість суттєвих.
E	60 – 69 (достатньо)	4			Здобувач освіти володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні.
FX	35 – 59 (незадовільно)	3	незадовільно	Початковий рівень	Здобувач освіти володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.
F	1 – 34 (незадовільно)	2			Здобувач освіти володіє матеріалом на елементарному рівні засвоєння, викладає його уривчастими реченнями, виявляє здатність викласти думку на елементарному рівні.
		1			Учень володіє навчальним матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів, що позначаються учнем окремими словами чи реченнями.

8. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ КУРСУ

1. Підручники та посібники.
2. Конспекти лекцій.
3. Лекції в електронному текстовому і графічному форматі.
4. Презентації.
5. Методичні вказівки до лабораторних робіт.
6. Матеріали для самостійного вивчення.
7. Індивідуальні завдання студентів.
8. Методичні вказівки до курсового проєкту.
9. Матеріали з контролю знань студентів
10. Стенди та інші наглядне обладнання аудиторії.

Вивчення дисципліни здобувачами освіти передбачає вміння використовувати різні інформаційні ресурси – опубліковану українську та іноземну літературу (нормативні документи, підручники, навчальні посібники, наукові періодичні та монографічні видання, словники, довідники тощо), методичну літературу та Інтернет-джерела.

9. РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Базова література

1. Коваль А. Введення в об'єктно-орієнтоване програмування. – Львів: ЛНУ, 2019. – 300 с.
2. Дмитренко А.В. Базові принципи ООП: реалізація на С#. – Київ: Вільямс, 2020. – 400 с.
3. Рижков І.В. Консольні програми на С#: теорія та практика. – Київ: КНУ, 2019. – 280 с.
4. Лівінський І.І., Бурлака М.С. Об'єктно-орієнтоване програмування на С#: навчальний посібник. – Київ: ЛНУ, 2022. – 350 с.
5. Воловик П.А. Об'єктно-орієнтоване програмування на С#: основи та розробка застосунків. – Львів: ЛНУ, 2020. – 450 с.
6. Смирнов В. Об'єктно-орієнтоване програмування: концепції, мова С# та застосування. – Харків: ХНУ, 2021. – 420 с.
7. Петрова О.В. Основи ООП з прикладами на С#. – Одеса: ОНУ, 2021. – 400 с.
8. Гнатюк А.С. ООП у середовищі .NET Framework: теорія і практика. – Київ: Університет «Україна», 2020. – 380 с.
9. Степанов О.М. Сучасні технології програмування: ООП з використанням С#. – Київ: КНУ, 2020. – 520 с.
10. Мальцев А.А. Віконні додатки у С# з використанням Windows Forms. – Київ: НаУКМА, 2019. – 320 с.

11. Антонюк Р.М. Проектування програмних систем на основі ООП. – Київ: КПІ, 2018. – 420 с.
12. Сорока М.В. Основи ООП: зразки коду на С#. – Київ: БХВ-Петербург, 2019. – 250 с.
13. Орлов В.М. Моделювання об'єктів та класів в ООП. – Київ: КПІ, 2018. – 310 с.
14. Хоменко А.П. С#: програмування з основами ООП. – Одеса: ОНУ, 2019. – 380 с.
15. Петров В.С. Windows Forms: створення віконних додатків на С#. – Київ: Університет «Україна», 2019. – 400 с.
16. Борисов М.А. Основи архітектури об'єктно-орієнтованих програмних систем. – Харків: ХНУ, 2021. – 360 с.
17. Кравченко О.П. Інтерфейси та поліморфізм у С#. – Київ: Вільямс, 2020. – 280 с.
18. Мельник І.Г. Інкапсуляція та наслідування в С#. – Київ: Вільямс, 2018. – 350 с.
19. Тарасенко О.В. Об'єктно-орієнтоване програмування для початківців. – Харків: ВНУ, 2018. – 300 с.
20. Соловійов М. Windows Forms: практичний курс з програмування у С#. – Київ: НаУКМА, 2019. – 290 с.

Допоміжна література

21. Злотник М.П. Робота з класами та об'єктами у С#. – Київ: Вільямс, 2021. – 250 с.
22. Федоренко Ю.В. Патерни проектування в об'єктно-орієнтованому програмуванні. – Львів: ЛНУ, 2021. – 420 с.
23. Бабич І.В. Windows Forms для розробників: покроковий гід. – Київ: Пітер, 2020. – 280 с.
24. Вербовська А.О. Консольні застосунки у С#: основи та практики. – Київ: Вільямс, 2020. – 320 с.
25. Ковальчук П.С. Поліморфізм та інтерфейси у С#. – Київ: БХВ-Петербург, 2019. – 300 с.
26. Степаненко М.М. Використання LINQ у С#. – Київ: НаУКМА, 2019. – 260 с.
27. Орлов В.М. Моделювання об'єктів та класів в ООП. – Київ: КПІ, 2018. – 310 с.
28. Борщевський А.Г. Управління помилками та виключення у С#. – Харків: ХНУ, 2018. – 240 с.
29. Кирилюк Д. О. Патерни ООП у С#. – Львів: ЛНУ, 2018. – 350 с.

Інформаційні ресурси

30. Офіційна документація С# [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/>
31. Підручник з Windows Forms [Електронний ресурс]. – Режим доступу:

- <https://docs.microsoft.com/en-us/dotnet/desktop/winforms>
- 32.Офіційна документація з Windows Presentation Foundation (WPF) [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>
 - 33.Офіційна документація ASP.NET Core для створення API [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/web-api/>
 - 34.Microsoft .NET: Мережеве програмування [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/framework/network-programming/>
 - 35.Офіційна документація про патерни проєктування [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/architecture/grpc-for-wcf-developers/design-patterns>
 - 36.Підручник з HTTP-клієнта в C# [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/tutorials/console-webapiclient>
 - 37.Офіційна документація по бібліотеці REST API в .NET [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/data-access/architect-modern-web-api>

Перелік деяких корисних ресурсів для самоосвіти

Електронні підручники, курси:

1. GitHub – репозиторії з прикладами – <https://github.com/topics/oop>
2. Практичні завдання з ООП – <https://www.codewars.com/>
3. Курс “Програмування на C#” (Prometheus) – <https://prometheus.org.ua/>
4. Безкоштовний курс з ООП (Coursera) – <https://www.coursera.org/specializations/object-oriented-programming>
5. Книга “C# для початківців” (PDF) – <https://books.br.com.ua/csharp>