

ОБ'ЄКТНО-ОРІЄНТОВАНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	-
Кредити ЄКТС	8,0
Форми навчання, для яких викладається дисципліна	Денна

Студент, який успішно завершив вивчення дисципліни, повинен: *застосовувати* знання у практичних ситуаціях, *адаптуватися* в умовах частотої зміни технологій професійної діяльності; *вміти* самостійно опановувати нові методи та технології розробки інформаційних систем; *використовувати* прикладні системи програмування, *розробляти* складні програмні комплекси з функціями захисту даних із застосуванням мови С#; *визначати* функціональні вимоги до розроблюваної програми, *розробляти* набори тестових даних та *давати оцінку* результативності якості прийнятих рішень; *застосовувати* методи та засоби криптографічного та технічного захисту інформації на об'єктах інформаційної діяльності, *вирішувати* задачі захисту інформації, що обробляється в інформаційно-телекомунікаційних системах, з використанням сучасних методів та засобів криптографічного захисту інформації; *використовувати* програмні та програмно-апаратні комплекси засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

Зміст навчальної дисципліни. Основні поняття С#, програмні рішення типових задач з використанням мови С#, об'єкти, класи, наслідування, перевантаження, поліморфізм, опрацювання виняткових ситуацій, потоки. Мета-класи, делегування, шаблони. Застосування принципів об'єктно-орієнтованого програмування. Об'єктно-орієнтований підхід до розробки і реалізації прикладних програмних систем, застосування об'єктно-орієнтованого підходу на всіх етапах життєвого циклу прикладної програмної системи, аналіз вимог до програмної системи і її попереднього проектування, реалізація, тестування і наступний супровід.

Пререквізити: -

Кореквізити: -

Запланована навчальна діяльність: всього 240 год., у т.ч. лекцій – 17 год., лабораторних занять – 34 год., практичних занять – 34 год., самостійної роботи – 155 год.

Форми (методи) навчання: пояснювально-ілюстративні, практичні, продуктивні та репродуктивні, тренінгові, застосування інформаційно-комп'ютерних технологій (MS Visual studio, Eclipse, ASP.NET тощо).

Форми оцінювання результатів навчання: усне опитування, тестування, захист лабораторних робіт, вирішення практичних завдань.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Griffiths Ian, Programming C# 8.0: Build Cloud, Web, and Desktop Applications/ Ian Griffiths. – O'Reilly, 2020. – 800с.
2. Price Mark, C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development / Mark Price. – Print2print, 2021. – 816р.
3. Perkins, B. Beginning C# 7 Programming with Visual Studio 2022/ B.Perkins, J.V.Hammer, J.D.Reid –John Wiley & Sons, Inc, 2018 – 912pp.
4. Albahari Joseph. C# 8.0 in a Nutshell: The Definitive Reference/Joseph Albahari, Eric Johanssen – O'Reilly Media, 2020г. – 1104с.
5. Мартін, Роберт С. Чистий код. Створення, аналіз і рефакторинг./ Роберт С. Мартін – Фабула, 2019р. – 416 с.
6. Troelsen, Andrew. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming / Andrew Troelsen, Phil Japikse. - Apress Media LLC, 2022. — 1705 p.
7. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnmu.edu.ua/>.
8. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnmu.edu.ua/>.

Викладачі: канд. техн. наук, доцент Джулій В.М.

ВСТУП

Дисципліна "Об'єктно-орієнтовані технології програмування" є етапом підготовки до самостійної практичної діяльності з розробки і експлуатації безпечних програмних додатків і тому займає провідне місце у підготовці бакалаврів з кібербезпеки.

Мета дисципліни: формування базових професійних компетенцій з розробки програмного забезпечення мовою програмування високого рівня та оволодіння студентами основними принципами і методами об'єктно-орієнтованого програмування, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності. Завдання, вирішення яких забезпечує досягнення мети: знання сучасних технологій програмування (об'єктно-орієнтоване, подійне-керування); освоєння принципів проектування алгоритмів завдань; знання технології розробки алгоритмів і програм; знання методів налагодження і тестування програм; знання методів розв'язання задач на ЕОМ в різних режимах; вміння забезпечити захист програмного забезпечення від копіювання та розповсюдження; вміння забезпечити цілісність та конфіденційність інформації при передачі по мережі; вміння ставити завдання і розробляти алгоритм її вирішення; вміння розробляти основні програмні документи; вміння використовувати прикладні системи програмування; володіння навичками розробки та налагодження програм на одній з об'єктно-орієнтованих мов програмування; формування наукового світогляду майбутнього фахівця

Предмет дисципліни. Основи програмування мовою C# на прикладі компілятора Microsoft ©Visual Studio©. Сучасні інформаційно-комунікаційні технології, сучасне програмно-апаратне забезпечення інформаційно-комунікаційних технологій у природничій і загально-професійній галузях інформаційної та/або кібербезпеки.

Завдання дисципліни. Показати студентам сучасний стан методів, технологій розробки і проектування безпечного програмного забезпечення в середовищі електронно-обчислювальних систем і мереж мовою C#. Навчити застосовувати принципи об'єктно-орієнтованого та подійно-керованого програмування на прикладі компілятора Microsoft ©Visual Studio©. Забезпечити набуття компетентностей та досягнення результатів навчання:

компетентності:

ЗК 1. Здатність застосовувати знання у практичних ситуаціях.

ЗК 5. Здатність до пошуку, оброблення та аналізу інформації.

ФК 3. Здатність до використання програмних та програмно-апаратних комплексів засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

ФК 10. Здатність застосовувати методи та засоби криптографічного та технічного захисту інформації на об'єктах інформаційної діяльності.

результати навчання:

ПРН 1(5). Адаптуватися в умовах частої зміни технологій професійної діяльності, прогнозувати кінцевий результат.

ПРН 2(47). Вирішувати задачі захисту інформації, що обробляється в інформаційно-телекомунікаційних системах, з використанням сучасних методів та засобів криптографічного захисту інформації.

ПРН 3(53). Вирішувати задачі аналізу програмного коду на наявність можливих загроз.

ПРН 4. Використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності.

ПРН 5. Самостійно опанувати нові методи та технології розробки інформаційних систем.

ПРН 6. Використовувати прикладні системи програмування, розробляти складні програмні комплекси з функціями захисту даних (із застосуванням мови C# тощо).

ПРН 7. Визначати функціональні вимоги до розроблюваної програми, розробляти набори тестових даних та давати оцінку результативності якості прийнятих рішень.

Студент, який успішно завершив вивчення дисципліни, повинен: *застосовувати* знання у практичних ситуаціях, *адаптуватися* в умовах частої зміни технологій професійної діяльності; *вміти* самостійно опановувати нові методи та технології розробки інформаційних систем; *використовувати* прикладні системи програмування, *розробляти* складні програмні комплекси з функціями захисту даних із застосуванням мови С#; *визначати* функціональні вимоги до розроблюваної програми, *розробляти* набори тестових даних та *давати оцінку* результативності якості прийнятих рішень; *застосовувати* методи та засоби криптографічного та технічного захисту інформації на об'єктах інформаційної діяльності, *вирішувати* задачі захисту інформації, що обробляється в інформаційно-телекомунікаційних системах, з використанням сучасних методів та засобів криптографічного захисту інформації; *використовувати* програмні та програмно-апаратні комплекси засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

СТРУКТУРА ЗАЛКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	лекції	практичні роботи	лабораторні роботи	самостійну роботу
Тема1. Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП. Основні поняття ООП.	4	4	6	24
Тема 2. Основні поняття мови програмування С#. Типізація в.NET	4	6	4	20
Тема 3. Концепція інкапсуляції і її реалізація в мові С#.	2	8	6	30
Тема 4. Концепція наслідування і її реалізація в мові С#.	3 (4/2)*	8	6	31 (30/32)*
Тема5. Концепція поліморфізму і її реалізація в мові С#	2	4	4	20
Тема 6. Подійно-кероване програмування в.NET. Файли.	2	4	8	30
Разом за семестр:	17 (18/16)*	34	34	155 (154/156)*

* За чисельником / за знаменником (розрахунок здійснюється відповідно до розкладу занять)

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
Тема 1. Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП. Основні поняття ООП		
1	<p>Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП. Сучасні підходи до програмування. Особливості декларативного підходу. Особливості імперативного підходу. Особливості функціонального підходу. Об'єктно-орієнтований підхід до програмування. Абстракція, інкапсуляція, спадкування й поліморфізм. Переваги й недоліки ООП. .NET як концепція. .NET як обчислювальна модель. .NET як технологічна платформа. .NET як інструментальний засіб. Common Language Runtime і .NET Framework.</p> <p>Літ.: [4] с. 31-147, с.257-323; [6] с. 47 – 193</p>	2
2	<p>Об'єктно-орієнтований підхід до програмування (продовження). Система типів Common Type System в.NET. Веб-сервіси в.NET. Компонентне програмування в.NET. Порівняння компонентного програмування з ООП. Переваги й недоліки .NET. Основні поняття мови програмування С#. Типізація в.NET. Еволюція мови програмування С#. Місце С# у сімействі мов програмування. Зіставлення мови С# з іншими мовами ООП (С++ і Java). Основні можливості мови програмування С#. Структура програми мовою С#. Синтаксис основних конструкцій мови С#. Переваги й недоліки мови С#. Основні поняття мови програмування С#. Неформальне й формальне визначення типів. Переваги теорій з типами. Класифікація систем типізації. Система типів (Common Type System, CTS) в.NET.</p> <p>Літ.: [4] с. 141-169; [6] с. 193 – 280</p>	2
Тема 2. Основні поняття мови програмування С#. Типізація в.NET		
3	<p>Основні поняття мови програмування С#. Базисні типи мови С# і їх відображення в CTS. Типи-Значення й посилальні типи; механізм (un)boxing. Простору імен. Перетворення типів в.NET [2-4]</p> <p>Робота з рядками. Рядки і клас System.String. Основні методи рядків. Форматування і інтерполяція рядків</p> <p>Літ.: [4] с. 429-453; [6] с. 97 – 280</p>	2
4	<p>Основні поняття ООП. Операції введення-виведення Інтуїтивні визначення об'єктів, класів і методів. Співвідношення понять об'єкта й класу. Концептуальна модель для класів і об'єктів. [3,5]</p> <p>Концепція інкапсуляції і її реалізація в мові С#. Поняття інкапсуляції в computer science. Інкапсуляція в ООП. Приклади інкапсуляції в мові С# (опис і застосування). Види областей видимості об'єктів. Рекомендації з розмежування областей видимості. Переваги інкапсуляції.</p> <p>Літ.: [4] с. 155-208; [6] с. 341 – 452</p>	2

Тема 3. Концепція інкапсуляції і її реалізація в мові С#.		
5	Типи значень і посилабельні типи Копіювання значень Посилабельні типи всередині типів значень Об'єкти класів як параметри методів. Константи і поля для читання Властивості і інкапсуляція Автоматичні властивості Ієрархія класів в.NET. Відображення класів .NET у типи мов SML і С#. Літ.: [4] с. 349-375; [6] с. 193 – 417	2
Тема 4. Концепція наслідування і її реалізація в мові С#.		
6	Наслідування. Конструктори в похідних класах. Перевизначення віртуальних методів. Заборона перевизначення методів. Приховування методів. Раннє і пізнє зв'язування. Абстрактні класи Абстрактні члени класів. Клас System.Object і його методи. Перетворення типів. Узагальнення. Обмеження універсальних типів. Використання декількох універсальних параметрів. Обмеження методів. Наслідування. Узагальнених типів. Значення null і Nullable-типи. Перетворення типів Nullable Літ.: [4] с. 403 – 429, с. 519 - 541; [6] с. 738 – 836	2 (2/1)*
7	Інтерфейси в перетвореннях типів Інтерфейси. Введення в інтерфейси Спадкування інтерфейсів Зміна реалізації інтерфейсів в похідних класах. Явне застосування інтерфейсів . Інтерфейси в узагальненнях. Узагальнені інтерфейси. Клонування об'єктів. Інтерфейс ICloneable. Поверхнєве копіювання Глибоке копіювання. Сортування об'єктів. Інтерфейс IComparable. Коваріантність і контраваріантних узагальнених інтерфейсів. Літ.: [4] с. 403 – 429, с.519 - 541; [6] с. 738 – 836	1 (2/1)*
Тема5. Концепція поліморфізму і її реалізація в мові С#		
8	Делегати, події. Визначення делегатів. Делегати як параметри методів. Застосування делегатів. Функціонал методів зворотного виклику. Події. Методи моделювання подій. Фрейми й функції як моделі подій. Літ.: [4] с. 501 - 591; [6] с. 738 – 836	2
Тема 6. Подійно-кероване програмування в.NET. Файли		
9	Робота з файлами. Класи File і FileInfo. Читання і запис файлу. клас FileStream. Довільний доступ до файлів. Закриття потоку. Літ.: [4] с. 501 - 591; [6] с. 738 – 836	2
Разом за семестр:		17 (18/16)*

* За чисельником / за знаменником (розрахунок здійснюється відповідно до розкладу занять)

Перелік лабораторних занять

№ п/п	Тема лабораторного заняття	Кільк. годин
1	Класи та об'єкти, інкапсуляція, наслідування Літ.: [4] с. 349-403; [6] с. 97 – 193	4
2	Поліморфізм. Механізми реалізації поліморфізму C#. Використання інтерфейсів Літ.: [4] с. 375-429; [6] с. 193 – 280	4
3	Робота з файлами. Основні принципи та прийоми розробки додатків, які використовують файли для зберігання даних. Літ.: [4] с. 31-147, с.257-323; [6] с. 47 – 193	4
4	Розширені можливості програмування на C#. Інтерфейси та делегати. Літ.: [4] с. 429-453; [6] с. 97 – 280	4
5	Подійно-орієнтоване програмування. Механізми обробки подій та виняткові ситуації. Літ.: [4] с. 225-245; [6] с. 836 – 937	4
6	Абстрагування та інкапсуляції на прикладі створення класу роботи з раціональними числами. Метод узагальнення Літ.: [4] с. 375-429; [6] с. 193 – 280	4
7	Класи – колекції Використання словників для створення телефонної книги Літ.: [4] с. 17-108; [6] с. 341 – 379	4
8	Класи – колекції Мова запитів LINQ на прикладі програми «Магазин» Літ.: [4] с. 17-108; [6] с. 341 – 379	4
9	Підсумкове заняття	2
Разом за семестр:		34

Перелік практичних занять

№ п/п	Тема практичного заняття	Кільк. годин
1	Масиви. Типи. Рядки. Метод Main() і аргументи командного рядка. Масиви. Типи. Рядки. Оператори та вирази в C#. Вивчення основних операцій з файлами мови C# Літ.: [4] с. 31-147, с.257-323; [6] с. 47 – 193	2
2	Класи та структури. Об'єкти. Екземпляри структури, класу. Ідентифікація об'єктів і рівність значень. Члени. Статичні члени. Літ.: [4] с. 349-403; [6] с. 97 – 193	2
3	Передача за посиланням та передача за значенням. Конструктори. Конструктори екземплярів. Закриті конструктори. Статичні конструктори. Створення конструктора копії. Деструктори. Літ.: [4] с. 141-169; [6] с. 193 – 280	2
4	Наслідування класів. Абстрактні та віртуальні методи. Інтерфейси. Поліморфізм в C#. Доступ до віртуальних членів базового класу з похідних класів Літ.: [4] с. 375-429; [6] с. 193 – 280	2
5	Індексатори. Делегати. Анонімні методи. Делегати з іменованими методами. Перевантаження операторів. Літ.: [4] с. 429-453; [6] с. 97 – 280	2
6	Універсальні типи. Універсальні типи під час виконання. Універсальні делегати. System.Collections.Generic - простір імен. IList<T> - інтерфейс. Літ.: [4] с. 429-453; [6] с. 193 – 280	2
7	Лямбда-вирази. Ітератори. Лямбди операторів. Область дії змінної в лямбда-виразах. Анонімні методи. Ітератори. Виключні ситуації. Літ.: [4] с. 155-208; [6] с. 341 – 452	2
8	Обробка подій. Робота з формами та компонентами (Windows Form Application) Літ.: [4] с. 225-245; [6] с. 836 – 937	2
9	Обробка подій з використанням графіки. Робота з формами та компонентами з використанням графіки (Windows Form Application) Літ.: [4] с. 247-257; [6] с. 341 – 379	2
10	Перевизначення віртуальних методів. Перевизначення віртуальних методів в узагальненому класі Коваріантність і контраваріантність делегатів. Літ.: [4] с. 375-403; [6] с. 379 – 452	2
11	Коваріантність і контраваріантність. Коваріантність і контраваріантність в узагальненнях. Використання коваріантності і контраваріантності інтерфейсів Літ.: [4] с. 349-375; [6] с. 193 – 417	2
12	Двоб'язний список. Двоб'язний список LinkedList <T>. Черга Queue <T>. Колекція Stack<T>. Колекція Dictionary<T, V>. Ініціалізація словників. Клас ObservableCollection. Літ.: [4] с. 17-108; [6] с. 341 – 379	2
13	Робота з дисками. Робота з файловою системою. Клас DriveInfo. Робота з каталогами. Клас Directory і DirectoryInfo. Літ.: [4] с. 403 – 429, с.519 - 541; [6] с. 738 – 836	2

14	Робота з файлами – загальні прийоми. Класи File і FileInfo. Читання і запис файлу. Клас FileStream. Літ.: [4] с. 519 - 541; [6] с. 738 – 836	2
15	Робота з текстовими і бінарними файлами. Створення та читання стислих файлів. Читання та запис текстових файлів. StreamReader і StreamWriter. Робота з бінарними файлами. BinaryWriter и BinaryReader. GZipStream і DeflateStream. Літ.: [4] с. 501 - 591; [6] с. 738 – 836	2
16	Серіалізація Введення в серіалізацію об'єктів. Бінарна серіалізація. BinaryFormatter. Серіалізація в XML. XmlSerializer. Літ.: [4] с. 501 - 591; [6] с. 738 – 836	2
17	Підсумкове заняття	2
Разом за семестр:		34

ЗМІСТ САМОСТІЙНОЇ (ІНДИВІДУАЛЬНОЇ) РОБОТИ

Об'єм самостійної роботи становить 155 годин. Він включає опрацювання лекційного матеріалу, підготовку до виконання лабораторних робіт і їх захисту, підготовку до поточного контролю, а також самостійну роботу студентів.

Номер тижня	Теми самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №1.	4
2	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №1.	6
3	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2	4
4	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 2	10
5	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 3.	10
6	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 3.	11 (11/12)*
7	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №4.	11 (10/11)*
8	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №4.	10
9	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 5	10
10	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №5	10
11	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 6.	10
12	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №6.	10
13	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №7.	10
14	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №7	10
15	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 8	10
16	Опрацювання лекційного матеріалу, Підготовка до захисту лабораторної роботи № 8	10
17	Опрацювання лекційного матеріалу. Підготовка до тестування.	10
Разом за семестр:		155 (154/156)*

* За чисельником / за знаменником (розрахунок здійснюється відповідно до розкладу аудиторних занять)

ТЕХНОЛОГІЇ ТА МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів, зокрема: лекції проводяться з використанням пояснювально-ілюстративних методів, лабораторні роботи та практичні заняття проводяться з використанням практичних, продуктивних, репродуктивних, тренінгових методів та з застосуванням інформаційно-комп'ютерних технологій (MS Visual studio, Eclipse, ASP.NET тощо).

Застосовувані при викладанні дисципліни методи навчання сприяють розвитку у студентів навичок soft skills: виконання частини лабораторних робіт передбачає роботу у малих групах з призначенням тим-лідера, що сприяє розвитку лідерських якостей у студентів, здатності до спілкування і організації командної роботи над спільними задачами, а змінюваність складу робочих груп між лабораторними роботами сприяє розвитку навичок адаптованості, гнучкості, комунікативності і оперативного налагоджування міжособистісних відносин в різних колективах; прилюдні захисти лабораторних робіт і виступи під час практичних занять з обґрунтуванням прийнятих рішень щодо вибору методів рішення завдань в діалозі з викладачем і групою сприяють формуванню і удосконаленню вмінь публічних виступів, емпатичного слухання, відстоювання власної точки зору, самоаналізу і самокритики; адаптованість, вміння користуватися інтернет-ресурсами та іншими джерелами інформації, синтезувати та критично осмислювати інформацію з різних джерел передбачені специфікою дисципліни, що передбачає рішення проблемних завдань із застосуванням творчих підходів в синтезі і аналізі програмних рішень і орієнтацію на роботу з постійно оновлюваними технологіями програмування; обмежений час на виконання лабораторних робіт, практичних і тестових завдань, чітко визначені терміни проходження контрольних точок і відпрацювання заборгованостей сприяють розвитку пунктуальності, здатності до самоорганізації та управління часом (тайм-менеджменту).

При вивченні дисципліни можуть бути захищені результати навчання, здобуті у неформальній освіті. Визнання результатів навчання, здобутих у неформальній освіті, реалізується згідно з чинним законодавством і регулюється Положенням про порядок перезарахування результатів навчання та визначення академічної різниці у ХНУ <https://www.khnu.km.ua/root/files/01/10/03/006.pdf>.

МЕТОДИ КОНТРОЛЮ

Поточний контроль здійснюється під час практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни.

При цьому використовуються такі методи поточного контролю:

- усне опитування;
- захист лабораторних робіт;
- вирішення практичних завдань;
- тестування.

Семестровий контроль проводиться у формі заліку. При виведенні підсумкової семестрової оцінки враховуються результати поточного контролю (залік за рейтингом формується автоматично за результатами поточного контролю).

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ

Оцінювання академічних досягнень студента здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за інституційною чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з урахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих видів її робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів за ваговими коефіцієнтами

Аудиторна робота		Контрольні заходи	Підсумковий контрольний захід
Лабораторні роботи №:	Практичні роботи №:	Тестовий контроль:	Семестровий контроль (залік)
1 - 8	1 - 17	Т 1-6	Залік за рейтингом
ВК:	0,4	0,2	

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт;

Оцінювання лабораторних робіт. Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення; вільне володіння студентом спеціальною термінологією і уміння фахово обґрунтувати прийняті конструктивні рішення; своєчасний захист лабораторної роботи.

Термін захисту звіту з лабораторної роботи вважається своєчасним, якщо студент захистив її в день виконання або на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Оцінку за лабораторне заняття викладач оголошує одразу після захисту звіту з лабораторної роботи і проставляє в електронний журнал дисципліни.

Оцінювання практичних занять. Оцінка, яка виставляється за практичне заняття, складається з таких елементів: здатність обрати оптимальний спосіб рішення завдання і обґрунтувати зроблений вибір; правильність та самостійність розв'язування задач, якість отримуваних результатів; вільне володіння студентом спеціальною термінологією і застосовуваними методами дисципліни; уміння фахово обґрунтувати прийняті конструктивні та аналітичні рішення.

Оцінку, отриману на практичному занятті, викладач оголошує студенту одразу після його відповіді і проставляє в електронний журнал дисципліни.

Впродовж семестру студент має отримати на практичних заняттях щонайменше три позитивні оцінки, щоб виконати програму дисципліни.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту

Сума балів за тестове завдання	1-5	6-12	13-18	19-20
Оцінка за 4-ри бальною шкалою	2	3	4	5

На тестування відводиться 20 хвилин (для закритої форми тестів – по одній хвилині на кожне завдання). Правильні відповіді студент записує у талоні відповідей. При цьому усі графи для відповідей мають бути заповнені символами, що відповідають правильним, на погляд студента, відповідям. Через 20 хвилин студенти здають викладачу завдання з талонами відповідей.

Тестування студент може також пройти і в он-лайн режимі в модульному середовищі для навчання.

Оцінку за тестування викладач пропоставляє в електронний журнал дисципліни.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за наведеними в таблиці критеріями оцінювання знань.

Критерії оцінювання знань студентів

Оцінка за інституційною шкалою	Узагальнений критерій
Відмінно	Студент глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає, логічний виклад відповіді державною мовою (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними інструментами. Студент не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки. При відповіді допустив дві-три несуттєві похибки.
Добре	Студент виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом і фаховою термінологією, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання закономірностей тощо. Відповідь студента будується на основі самостійного мислення. Студент у відповіді допустив дві–три несуттєві помилки.
Задовільно	Студент виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент має слабкі знання структури курсу, допускає неточності і суттєві помилки у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно	Студент виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткової роботи з вивчення дисципліни.

Якщо студент отримав негативну оцінку за певним видом робіт, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю.

У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Студент, який у встановлені терміни не виконав індивідуальний план поточної роботи з дисципліни повністю або частково, до здачі підсумкового контрольного заходу не допускається.

Залік вважається зданим при отриманні студентом за зведеними результатами поточного контролю підсумкової оцінки з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться оцінка за двобальною шкалою, а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ECTS встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ECTS

Оцінка ECTS	Інституційна інтервальна шкала балів	Інституційна оцінка, критерії оцінювання		
A	4,75–5,00	5	Зараховано	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25–4,74	4		Добре – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4		Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3		Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3		Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незараховано	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2		Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ ЗДОБУТИХ СТУДЕНТАМИ РЕЗУЛЬТАТІВ НАВЧАННЯ

1. Загальні відомості про програми на C#. Структура проектів в C#.
2. Елементи мови. Службові слова. Імена. Коментарі.
3. Розміщення даних у пам'яті. Тип даних. Адресний вираз.
4. Функції. Область існування імені. Область видимості та простору імен.
5. Типи. Базові стандартні типи. Константи. Змінні.
6. Операції C#. Арифметичні операції. Операції присвоєння. Операції відношення та еквівалентності. Логічні операції.
7. Операції адресації та непрямой адресації.
8. Пріоритет і асоціативність операцій. Перегрузка операцій.
9. Масиви у C#. Операції з масивами. Правила роботи з масивами.
10. Структури та об'єднання. Структури в стилі C#.
11. Об'єднання. Об'єднання, що не мають імені. Доступ до членів-даних структури, об'єднання.
12. Екземпляри структури, об'єднання. Масиви структур.
13. Вказівники, зсилки та функції в C#.
14. Локальний та динамічний розподіл пам'яті.
15. Оператори new и delete. Динамічне розміщення об'єктів та вказівників.
16. Правила роботи з вказівниками. Розіменування вказівників.
17. Оператори new[], оператори delete[].
18. Функції в C#. Передача аргументів та повернення результату.
19. Застосування при передачі параметрів специфікації const. Параметри зі значеннями по умовчанню.
20. Перегрузка функцій. Приведення типів.
21. Функції-члени класу. Передача у функції змінного числа параметрів.
22. Вбудовані функції inline.
23. Шаблони функцій.
24. Область видимості функцій. Правила, що визначають область видимості.
25. Функція main(). Аргументи функції main. Аргументи командного рядка.
26. Основні поняття ООП. Інкапсуляція, поліморфізм, наслідування і засоби їх реалізації.
27. Об'єкт, властивості, методи, події.
28. Класи. Визначення класу. Ідентифікатори класу. Тіло класу.
29. Приховування інформації. Ієрархії класів та наслідування.
30. Інкапсуляція. Керування доступом до класу. Приватні, загальнодоступні, захищені.
31. Класи пам'яті для об'єктів класів. Область видимості класу. Порожні класи.
32. Вкладені класи. Правила доступу для вкладених класів.
33. Екземпляри класу. Використання членів даних. Статичні члени-данні.
34. Об'єкти класу як члени даних. Вказівники як члени даних.
35. Використання функцій-членів. Прості, статичні функції члени.
36. Конструктори. Деструктори.
37. Дружні функції. Властивості дружніх функцій.
38. Шаблони класів. Вкладені шаблонні класи.
39. Наслідування. Обмеження наслідування. в C#. Просте наслідування.
40. Специфікатори доступу базових класів. Класи для наслідування. Аргументи передані в базовий клас. Порядок виклику конструкторів. Порядок виклику деструкторів.
41. Перетворення типів у похідних класах. Дозвіл області видимості.
42. Множинне наслідування. Оголошення класу з декількома базовими класами. Виклик конструкторів базових класів.
43. Використання віртуальних базових класів. Використання віртуальних і невіртуальних базових класів разом.

44. Використання перетворення типів. Правила виклику функцій базових класів.
45. Використання дозволу видимості при множинному наслідуванні.
46. Поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Перевизначення функції.
47. Абстрактні класи. Обмеження віртуальних функцій. Віртуальні оператори.
48. Поліморфізм при простому наслідуванні. Поліморфізм при множинному наслідуванні.
49. Виклик поліморфних функцій базового класу. Віртуальні функції та ієрархії класифікацій. Виклик віртуальних функцій у конструкторах.
50. Потік C#. Потоки як узагальнені фільтри. Стандартний потоковий ввід-вивід класів користувача.
51. Потоки і файловий ввід-вивід. Використання текстових файлів для введення. Вивід текстових файлів.
52. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями. Методи опрацювання подій.
53. Шаблони функцій і класів. Основні властивості шаблонів класів. Компонентні функції параметризованих класів.
54. Стандартна бібліотека шаблонів. Класи-контейнери. Асоціативні контейнери.

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни „Об'єктно-орієнтовані технології програмування” повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою, розміщеною в електронному варіанті в модульному середовищі.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Griffiths Ian, Programming C# 8.0: Build Cloud, Web, and Desktop Applications/ Ian Griffiths. – O'Reilly, 2020. – 800с.
2. Price Mark, C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development / Mark Price. – Print2print, 2021. – 816р.
3. Perkins, B. Beginning C# 7 Programming with Visual Studio 2022/ B.Perkins, J.V.Hammer, J.D.Reid –John Wiley & Sons, Inc, 2018 – 912pp.
4. Albahari Joseph. C# 8.0 in a Nutshell: The Definitive Reference/Joseph Albahari, Eric Johanssen – O'Reilly Media, 2020г. – 1104с.
5. Мартін, Роберт С. Чистий код. Створення, аналіз і рефакторинг./ Роберт С. Мартін – Фабула, 2019р. – 416 с.
6. Troelsen, Andrew. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming / Andrew Troelsen, Phil Japikse. - Apress Media LLC, 2022. — 1705 p.
7. Richter, Jeffrey. Програмування на платформі Microsoft .NET Framework мовою C#/ Jeffrey Richter – Microsoft Press, 2021р. – 896с.
8. Bender, J. Professional Test-Driven Development with C#: Developing / J.Bender, J. McWherter – 2020.– 327pp.
9. Stellman Andrew. Head First C#, 4th Edition/ Andrew Stellman, Jennifer Greene – O'Reilly, 2020г. – 748 с.
10. Методи і алгоритми захисту інформаційних ресурсів комп'ютерних систем: навчальний посібник / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун. – Хмельницький: ХмНУ, 2020. – 196 с.
11. Джулій В.М. Модель формування цілей інформаційного забезпечення підтримки процесів надання знань / В.М. Джулій, І.В. Гурман, А.А. Маковей // Збірник наукових праць Військо-вого інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2019. – Вип. №63. – С. 90-98.
12. Cleary Stephen. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming, 2nd Edition / Stephen Cleary– O'Reilly Media, 2019г. – 372 с.

Додаткова

13. Джемс, Філіп. Мова програмування C# 7 та платформи .NET та .NET Core/ Філіп Джемс, Ендрю Троелсен – Діалектика, 2021р. – 1328с.
14. Робсон, Е. Патерни проектування./ Е. Робсон, Е. Фрімен – Фабула. 2020р. – 672 с.
15. Фрімен Адам, ASP.NET Core 3 з прикладами на C# для професіоналів /Адам Фрімен- Apress, 2021. - 1184с.
16. Olsson, M. C# 7 Quick Syntax Reference: A Pocket Guide to the Language, APIs, and Library, 2018 – 180pp.
17. Fowler, M. Refactoring: Improving the Design of Existing Code (2nd Edition) (Addison-Wesley Signature Series (Fowler))– 2018. – 448 pp.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua/>.
2. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnu.edu.ua/>