ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
Декан факультету ІТ
Савенко О.С.
1 вересня 2022 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване проектування

Назва

Статус дисципліни: вибіркова
Факультет – Факультет інформаційних технологій
Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
					Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС				
			Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття					Залік	Іспит
Д	2	парний	8.0	240	108	36	36	36		132	-	-	+	-
Разом ДФН			8.0	240	108	36	36	36		132			**1**	

Робоча програма складена на основі освітньо-професійної програми підготовки бакалаврів

Програма складена _____ Лисенком С.М.
Підпис Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол № 1 від 12 серпня 2022 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Говорущенко Т.О.
Підпис Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради _____ Савенко О.С.
Підпис Ініціали, прізвище

Хмельницький 2022

ВСТУП

Мета викладання дисципліни. Дисципліна “Об’єктно-орієнтоване проектування” є вибірковою з програмування та характеризується широким міждисциплінарним підходом.

Метою курсу є оволодіння студентами основними принципами проектування та розроблення програмного забезпечення на основі об’єктно-орієнтованого підходу мовою C#, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності. Таким чином, мета і завдання дисципліни “Об’єктно-орієнтоване проектування” навчити застосовувати принципи алгоритмічного та об'єктно-орієнтованого програмування на всіх етапах життєвого циклу прикладної програмної системи, починаючи з аналізу вимог до програмної системи і її попереднього проектування, і закінчуючи її реалізацією, тестуванням і наступним супроводом.

Предмет дисципліни. Основи програмування мовою C# на прикладі компілятора Microsoft© Visual Studio©.

Завдання дисципліни. Навчити застосовувати принципи алгоритмічного та об'єктно-орієнтованого програмування на прикладі компілятора Microsoft ©Visual Studio©; розглянути всі основні поняття C#, такі як об'єкти, класи, наслідування, перевантаження, поліморфізм, опрацювання виняткових ситуацій, потоки; розглянути такі специфічні питання, як мета-класи, делегування, шаблони; навчити процедурному та об'єктно-орієнтованому підходах до розробки і реалізації прикладних програмних систем; довести доцільність і плодотворність систематичного застосування процедурного та об'єктно-орієнтованого підходів на всіх етапах життєвого циклу прикладної програмної системи.

Після вивчення дисципліни **"Об’єктно-орієнтоване проектування"** студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

знати:

- об’єкт, предмет, задачі, проблематику дисципліни та її основні розділи;
- наукові і математичні положення, що лежать в основі функціонування програмних систем; базові поняття й визначення, використовувані у галузі програмної інженерії; основи проектування програмного забезпечення мовою C#;
- інновації у галузі програмної інженерії

***уміти*:**

- застосовувати основи комп’ютерної інженерії для розв’язування задач синтезу програмних систем на основі об’єктно-орієнтованого підходу із застосуванням мови програмування C#;
- системно мислити та застосовувати творчі здібності до формування принципово нових ідей щодо розроблення програмних систем засобами ООП мовою програмування мовою C#;
- застосовувати знання принципів об’єктно-орієнтованого програмування мовою C# для ідентифікації, формулювання і розв’язування технічних задач спеціальності;
- вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем застосовуючи об’єктно-орієнтований підхід до програмування мовою C#;
- застосовувати теорію та методи системного аналізу, математичного і комп’ютерного моделювання при проектуванні та розробленні програмного забезпечення та програмних систем на основі об’єктно-орієнтованого підходу мовою C#;
- оцінювати результати роботи, захищати, пояснювати та аргументувати розроблену програмну систему мовою C#, отримані результати;
- вміти здійснювати пошук інформації в різних джерелах для розв’язання задач побудови ПЗ на основі об’єктно-орієнтованого підходу програмування мовою C#;

бути здатним:

- розв’язувати складні задачі і проблеми в галузі проектування програмного забезпечення мовою C#, що передбачає проведення досліджень та/або здійснення інновацій;
- абстрактно мислити, аналізувати і синтезувати програмні системи на основі об'єктно-орієнтованого підходу програмування мовою C#;
- застосовувати знання принципів об’єктно-орієнтованого програмування на практиці;
- розробляти програмне забезпечення, програмні компоненти комп’ютерних систем та мереж, Інтернет додатки, кіберфізичних систем з використанням об’єктно-орієнтованих методів програмування мовою C#;

- створювати системне та прикладне програмне забезпечення комп’ютерних систем та мереж із застосуванням об'єктно-орієнтованого підходу програмування мовою C#;
- проектувати програмні системи та їхні компоненти із застосуванням об’єктно-орієнтованих методів програмування мовою C# з урахуванням усіх аспектів їх життєвого циклу ПЗ, включаючи створення, налаштування, експлуатацію та обслуговування.

ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОЕКТУВАННЯ

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

Результати навчання

Студент, який успішно завершив вивчення дисципліни, повинен: використовувати сучасні методи і мови програмування для розроблення алгоритмічного та об’єктно-орієнтованого програмного забезпечення; створювати системне та прикладне програмне забезпечення комп’ютерних систем та мереж мовою C#; забезпечувати проектування та розроблення якісних програмних засобів; розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем; використовувати навички програмування, технології розроблення алгоритмів і комп’ютерних програм мовами високого рівня із застосуванням об’єктно-орієнтованого програмування для розв’язання задач проектування і використання програмно-технічних засобів комп’ютерних систем та мереж.

Зміст навчальної дисципліни. Застосування принципів процедурного та об'єктно-орієнтованого програмування. Основні поняття C#, об'єкти, класи, наслідування, перевантаження, поліморфізм, опрацювання виняткових ситуацій, потоки. Мета-класи, делегування, шаблони. Об'єктно-орієнтований підхід до розробки і реалізації прикладних програмних систем, доцільність і плодотворність систематичного 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.

Запланована навчальна діяльність: лекції – 36 год., практичні заняття – 36 год., лабораторні заняття – 36 год., самостійна робота – 132 год.; разом – 240 год

Методи навчання: словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, частково-пошукові (практичні та лабораторні заняття), практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми оцінювання результатів навчання: захист лабораторних та практичних робіт, тестовий контроль.
Форма семестрового контролю: залік

Навчальні ресурси:

1. Технічна документація ASP.NET Core Microsoft. https://docs.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-6.0
2. Павлишко А.В. Савєльєва О.В. алгоритмізація та програмування. Одеса .2017 с.123.
3. І. О. Завадський, Р. І. Заболотний , Основи візуального програмування. Київ Видавнича група BHV. С.271.
4. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою C# 7.0. Тернопіль 2017. с.302.
5. Joseph Albahari. C# 7.0 Pocket Reference: Instant Help for C# 7.0 Programmers 1st Edition. O'Reilly Media. 2017. С.240 (англ).
6. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.
7. Модульне середовище для навчання MOODLE. Доступ до ресурсу: https://msn.khnu.km.ua.

Викладач: доктор техн. наук, професор Лисенко С.М.

1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	**Кількість годин відведених на:**			
	Лекції	Лабораторні роботи	Практичні роботи	**СРС**
Четвертий семестр				
Тема 1. Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП.	4	4	4	14
Тема 2. Інкапсуляція. Наслідування. Поліморфізм	4	4	4	28
Тема 3. Делегати, події і лямбди	4	4	4	14
Тема 4. Колекції. Введення в колекції	4	4	4	14
Тема 5. Робота з потоками і файлової системою	4	4	4	14
Тема 6. Основи LINQ	4	4	4	8
Тема 7. Основи Entity Framework	4	4	4	32
Тема 8. Управління пам'яттю	8	8	8	8
Години	36	36	36	132
Разом	**240 (8.0 кредити)**			

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
2.1. Зміст лекційного курсу*

№ п/п	Перелік тем лекцій, їх анотація	Години
	Четвертий семестр	
1	**Тема 1.** Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП. ***Лекція 1.*** Загальна структура програми на C#. Масиви. Рядки. Читання з файлу та запис у файл. Класи та структури. Інкапсуляція.	2
2	**Тема 1.** Об'єктно-орієнтований підхід до програмування. Платформа .NET і її застосування для ООП. ***Лекція 2.*** Рядки. Оголошення та ініціалізація рядків. Правильні та буквальні рядкові літерали	2
3	**Тема 2.** Інкапсуляція. Наслідування. Поліморфізм ***Лекція 3.*** Введення в C#. Мова C# і платформа .NET Літ.: [1,5,7,8, 9, 10]	2
4	**Тема 2.** Інкапсуляція. Наслідування. Поліморфізм ***Лекція 4.*** Інкапсуляція. Константи і поля для читання Літ.: [3,4,5,8,11]	2
5	**Тема 2.** Інкапсуляція. Наслідування. Поліморфізм ***Лекція 5.*** Наслідування. Доступ до членів базового класу з класу-наслідника	2
6	**Тема 2.** Інкапсуляція. Наслідування. Поліморфізм ***Лекція 6.*** Перетворення типів Літ.: [1, 5, 9]	2
7	**Тема 3.** Делегати, події і лямбди ***Лекція 7.*** Делегати, події і лямбди Літ.: [2,3,4]	2
8	**Тема 3.** Делегати, події і лямбди ***Лекція 8***. Анонімні метод. Лямбда-вирази в обробці подій Літ.: [2,5,8]	2
9	**Тема 4.** Колекції. Введення в колекції ***Лекція 9.*** Колекції. Введення в колекції .Літ.: [1,2]	2
10	**Тема 4.** Колекції. Введення в колекції ***Лекція 10.*** Колекції. Введення в колекції Літ.: [4,5,8]	2
11	**Тема 5.** Робота з потоками і файлової системою ***Лекція 11.*** Робота з потоками і файлової системою Літ.: [1,2,4,7,8]	2
12	**Тема 5.** Робота з потоками і файлової системою ***Лекція 12.*** Серіалізація. Введення в серіалізацію об'єктів. Літ.: [1,9,10].	2
13	**Тема 6.** Основи LINQ ***Лекція 13.***Основи LINQ Літ.: [5,7].	
14	**Тема 7.** Основи Entity Framework ***Лекція 14.*** Введення в Entity Framework 6 Взаємодія з даними. Що таке Entity Framework. Перше додаток. Підхід Code First. Code First до існуючої бази даних. Угоди по найменуванню в Code First. Автоматизація Code First. Автоматизація Code First і EF Power Tools. Database First. Model First. Літ.: [5,7].	2
15	**Тема 7.** Основи Entity Framework ***Лекція 15***. Основи Entity Framework. Основні операції з даними. Операції з даними. практичний приклад. Рядок підключення. Навігаційні властивості і завантаження даних. Зв'язок один до одного. Зв'язок один до багатьох. Зв'язок один до багатьох. практичний приклад. Зв'язок багато до багатьох. Зв'язок багато до багатьох. практичний приклад.• Ініціалізація бази даних. Паралелізм в Entity Framework. Управління транзакціями.• Міграції. Generic Repository [1,2,6,7]	2
16	**Тема 7.** Основи Entity Framework ***Лекція 16.*** LINQ to Entities. Введення в LINQ to Entities. Вибірка і проекція з бази даних. Сортування. З'єднання таблиць. Угрупування. Операції з множинами: об'єднання, перетин, різниця. Агрегатні операції. IEnumerable і IQueryable. AsNoTracking Літ.: [3,4,8]	2

17	**Тема 7.** Основи Entity Framework ***Лекція 17*** Глава 4. SQL в Entity Framework. Робота з SQL. Збережені функції. Збережені процедури. Літ.: [4,6,7,8]	2
18	**Тема 8.** Управління пам'яттю ***Лекція 18.*** Збирання сміття, управління пам'яттю і вказівники. Збиральник сміття в C #. Літ.: [10-14]	2
	Разом	36

2.2 Зміст лабораторних занять

№ п/п	Теми лабораторних робіт	Кіль кість годин
Четвертий семестр		
1	***Лабораторна робота №1.*** Загальна структура програми на C#. Метод Main() і аргументи командного рядка. Масиви. Типи. Рядки. в Оператори та вирази в C#. Вивчення основних операцій з файлами мови C# [1,4].	4
2	***Лабораторна робота №2.*** Класи та структури. Об'єкти. Екземпляри структури і екземпляри класу. Ідентифікація об'єктів і рівність значень. Члени. Статичні члени. Властивості. Властивості, що реалізуються автоматично. Методи. Доступ до методів. Передача за посиланням та передача за значенням. Значення, що повертаються. Конструктори. Конструктори екземплярів. Закриті конструктори. Статичні конструктори. Створення конструктора копії. Деструктори. [1-4].	4
3	***Лабораторна робота №3.*** Наслідування класів. Модифікатор new. Ключове слово base. Модифікатор sealed. Ключове слово this. Абстрактні та віртуальні методи. Абстрактні та запаковані класи та члени класів. Запаковані класи та члени класів. Віртуальні методи. Інтерфейси. Явна реалізація інтерфейса. Поліморфізм в C#. Запобігання перевизначення віртуальних членів похідними класами. Доступ до віртуальних членів базового класу з похідних класів [2,6,7,10].	4
4	***Лабораторна робота №4.*** Індексатори. Індексатори в інтерфейсах. Порівняння властивостей та індексаторів. Делегати. Анонімні методи. Делегати з іменованими методами і делегати з анонімними методами. Перевантаження операторів. [2,6,7,10].	4
5	***Лабораторна робота №5.*** Введення в універсальні шаблони. Універсальні типи під час виконання. Обмеження параметрів типу. Застосування обмежень. Універсальні класи. Універсальні інтерфейси. Універсальні методи. Універсальні делегати. System.Collections.Generic - простір імен. IList<T> - інтерфейс. [1-4,10].	4
6	***Лабораторна робота № 6.*** Анонімні функції. Лямбда-вирази. Лямбди операторів. Лямбди зі стандартними операторами запитів. Виведення типу в лямбда-виразах. Область дії змінної в лямбда-виразах. Анонімні методи. Ітератори. Використання виключних ситуацій. Ключові слова try, catch и finally. [1,9,10].	4
7	***Лабораторна робота №7.*** Обробка подій. Робота з формами та компонентами (Windows Form Application) **[1,2,6,7].**	4
8	***Лабораторна робота №8.*** Обробка подій. Робота з формами та компонентами з використанням графіки (Windows Form Application) [6,9,10,11]	4
9	***Підсумкове заняття***	4
Всього		36

2.3 Зміст практичних занять

№ п/п	Теми практичних робіт	Кіль кість годин
Четвертий семестр		
1	***Практична робота №1.*** Загальна структура програми на C#. Метод Main() і аргументи командного рядка. Масиви. Типи. Рядки. в Оператори та вирази в C#. Вивчення основних операцій з файлами мови C# [1,4].	4
2	***Практична робота №2.*** Класи та структури. Об'єкти. Екземпляри структури і екземпляри класу. Ідентифікація об'єктів і рівність значень. Члени. Статичні члени. Властивості. Властивості, що реалізуються автоматично. Методи. Доступ до методів. Передача за посиланням та передача за значенням. Значення, що повертаються. Конструктори. Конструктори екземплярів. Закриті конструктори. Статичні конструктори. Створення конструктора копії. Деструктори. [1-4].	4
3	***Практична робота №3.*** Наслідування класів. Модифікатор new. Ключове слово base. Модифікатор sealed. Ключове слово this. Абстрактні та віртуальні методи. Абстрактні та запаковані класи та члени класів. Запаковані класи та члени класів. Віртуальні методи. Інтерфейси. Явна реалізація інтерфейса. Поліморфізм в C#. Запобігання перевизначення віртуальних членів похідними класами. Доступ до віртуальних членів базового класу з похідних класів [2,6,7,10].	4
4	***Практична робота №4.*** Індексатори. Індексатори в інтерфейсах. Порівняння властивостей та індексаторів. Делегати. Анонімні методи. Делегати з іменованими методами і делегати з анонімними методами. Перевантаження операторів. [2,6,7,10].	4
5	***Практична робота №5.*** Введення в універсальні шаблони. Універсальні типи під час виконання. Обмеження параметрів типу. Застосування обмежень. Універсальні класи. Універсальні інтерфейси. Універсальні методи. Універсальні делегати. System.Collections.Generic - простір імен. IList<T> - інтерфейс. [1-4,10].	4
6	***Практична робота № 6.*** Анонімні функції. Лямбда-вирази. Лямбди операторів. Лямбди зі стандартними операторами запитів. Виведення типу в лямбда-виразах. Область дії змінної в лямбда-виразах. Анонімні методи. Ітератори. Використання виключних ситуацій. Ключові слова try, catch и finally. [1,9,10].	4
7	***Практична робота №7.*** Обробка подій. Робота з формами та компонентами (Windows Form Application) **[1,2,6,7].**	4
8	***Практична робота №8.*** Обробка подій. Робота з формами та компонентами з використанням графіки (Windows Form Application) [6,9,10,11]	4
9	***Підсумкове заняття***	4
Всього		36

2.4 Зміст самостійної (індивідуальної) роботи

Об'єм самостійної роботи з дисципліни "Об'єктно-орієнтоване проектування" в IV семестрі становить 132 годин. Він включає опрацювання лекційного матеріалу, підготовку до виконання лабораторних робіт і їх захисту, підготовку до поточного контролю.

Керівництво самостійною роботою та виконанням курсового проекту здійснює викладач згідно з розкладом консультацій в позаурочний час

Номер тижня	Вид самостійної роботи	К-ть годин
1	Опрацювання лекційного матеріалу. Підготовка до ЛР1	7
2	Опрацювання лекційного матеріалу.	7
3	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР1. Підготовка до ЛР2	7
4	Опрацювання лекційного матеріалу.	7
5	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР2. Підготовка до ЛР3	7
6	Опрацювання лекційного матеріалу.	7
7	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР3. Підготовка до ЛР4	7
8	Опрацювання лекційного матеріалу.	7
9	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР4. Підготовка до ЛР5	7
10	Опрацювання лекційного матеріалу.	7
11	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР5. Підготовка до ЛР6	7
12	Опрацювання лекційного матеріалу.	7
13	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР6. Підготовка до ЛР7	8
14	Опрацювання лекційного матеріалу.	8
15	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР7. Підготовка до ЛР8	8
16	Опрацювання лекційного матеріалу.	8
17	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР8. Підготовка до ТК	8
18	Опрацювання лекційного матеріалу.	8
	Разом за семестр:	132

Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснює викладач згідно з розкладом консультацій у позаурочний час.

3. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, лекції проводяться в основному словесними методами, а лабораторні заняття проводяться з використанням інформаційних технологій, майстер-класів, практикумів і мають за мету – набуття студентами практичних навичок з проектування ПЗ за сучасними методиками, деталювання креслень, користування спеціальними конструкторськими інструментами тощо.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань предметної області, вміє розширити їх. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: проектувати програмне забезпечення із застосувати веб-орієнтовані технології програмування.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєнні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєнні знання на нетипові, нестандартні ситуації.

Формами і методами оцінювання результатів навчання є контрольна робота та захисти лабораторних робіт.

Кожний вид роботи з дисципліни оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: письмова контрольна робота, усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється під час проведення лабораторних занять; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Письмова контрольна робота проводиться на останньому практичному занятті і включає два теоретичні питання та одну практичну задачу.

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 протоколу і графічної частини; вміння студента обґрунтувати прийняті конструктивні рішення; своєчасний захист лабораторної роботи. Для виконання програми дисципліни студент повинен отримати вісім оцінок за лабораторні роботи в семестрі і написати підсумкову контрольну роботу на позитивну оцінку.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи з неповажної причини студент за позитивну відповідь отримує оцінку «задовільно».

Пропущене лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до закінчення теоретичних занять у семестрі.

При оцінюванні знань студентів філософії викладач керується такими критеріями.

Оцінка "відмінно" виставляється студенту, який глибоко засвоїв методи розв’язування практичних задач та вміє їх раціонально застосувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре” отримує студент за повне засвоєння навчального матеріалу, володіння

понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „добре” отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки "задовільно" заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок застосування методів розв’язування практичних задач, але припустився неточностей. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно" заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички застосування технологій розв’язування практичних задач.

Оцінка „незадовільно” виставляється, коли студент має 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 додаткових знань з курсу.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів *денної* форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота		Самостійна, індивідуальна робота	Семестровий контроль (залік)
V семестр			
Лабораторні роботи:	Практичні роботи:	Оцінювання проектів	
1-8	1-8	контрольна робота	за рейтингом
ВК: 0,4	0,2	0,4	0

Примітка: Т – тема дисципліни; ВК – ваговий коефіцієнт; КР – контрольна робота

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначать конкретну оцінку ECTS.

Оцінювання тестових завдань

Тематичний тест для кожного студента складається з двадцяти п’яти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 25. Оцінювання здійснюється за чотирибальною шкалою. Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–13	14–16	17–22	23–25
Оцінка	2	3	4	5

На тестування відводиться 25 хвилин (для закритої форми тестів – по одній хвилині на кожне завдання). Тестування студент проходить в он-лайн режимі в модульному середовищі MOODLE.

Якщо студент отримав негативну оцінку, 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”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у табл. 9.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	**Інституцій на шкала балів**	**Інституцій на 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.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ ФОРМ 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 непрямої 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-данних 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. Динамичне 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 члени класу.
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.
41. Порядок виклику конструкторів. Порядок виклику деструкторів.
42. Перетворення типів у похідних класах. Дозвіл області видимості.
43. Множинне наслідування. Оголошення класу з декількома базовими класами. Виклик конструкторів базових класів.
44. Використання віртуальних базових класів. Використання віртуальних і невіртуальних базових класів разом.
45. Використання перетворення типів. Правила виклику функцій базових класів.
46. Використання дозволу видимості при множинному наслідуванні.
47. Поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Перевизначення функції.
48. Абстрактні класи. Обмеження віртуальних функцій. Віртуальні оператори.
49. Поліморфізм при простому наслідуванні. Поліморфізм при множинному наслідуванні.
50. Виклик поліморфних функцій базового класу. Віртуальні функції та ієрархії класифікацій. Виклик віртуальних функцій у конструкторах.
51. Потік C#. Потоки як узагальнені фільтри. Стандартний потоковий ввід-вивід класів користувача.
52. Потоки і файловий ввід-вивід. Використання текстових файлів для введення. Вивід текстових файлів.
53. Обробка подій. Об’єктно-орієнтована програма, яка керована подіями. Методи опрацювання подій.
54. Шаблони функцій і класів. Основні властивості шаблонів класів. Компонентні функції параметризованих класів.
55. Стандартна бібліотека шаблонів. Класи-контейнери. Асоціативні контейнери. Алгоритми.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Об’єктно-орієнтоване проектування» забезпечений необхідною навчально-методичною літературою.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Технічна документація ASP.NET Core Microsoft. https://docs.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-6.0.
2. Дженніфер Грін, Ендрю Стеллман. Head First C#: Посібник для навчання з програмування в реальному світі за допомогою C# та .NET Core 4th Edition. O'Reilly Media. 2021 рік с.800 (англ).
3. Джозеф Альбахарі ,Бен Альбахарі.C# 8.0 Pocket Reference: миттєва допомога для програмістів C# 8.0, 1-е видання. O'Reilly Media. 2019 . с.248 (англ).
4. Джозеф Альбахарі, Ерік Йоганнсен. C# 8.0 in Nutshell: The Definitive Reference 1st Edition. O'Reilly Media. 2020. с. 1088(англ).
5. Ian Griffiths. Programming C# 8.0: Build Windows, Web, and Desktop Applications 1st Edition. O'Reilly Media. 2020. С.802 (англ).
6. Коноваленко І.В. Програмування мовою C# 6.0. Тернопіль, ТНТУ. 2016. С.227.
7. Д. В. Настенко, А. Б. Нестерко. ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ Частина 1. Основи об'єктно-орієнтованого програмування на мові C#. Київ НТУУ «КПІ» 2016 с.76.
8. Павлишко А.В. Савєльєва О.В. АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ. Одеса .2017 с.123.
9. І. О. ЗАВАДСЬКИЙ, Р. І. ЗАБОЛОТНИЙ , Основи візуального програмування. Київ Видавнича група BHV. С.271.
10. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою C# 7.0. Тернопіль 2017. с.302.
11. Joseph Albahari. C# 7.0 Pocket Reference: Instant Help for C# 7.0 Programmers 1st Edition. O'Reilly Media. 2017. С.240 (англ).
12. Joseph Albahari. C# 10 in a Nutshell. The Definitive Reference. O'Reilly Media.2022. с.1058 (англ)
13. Джозеф Альбахарі, Бен Альбахарі. Кишеньковий довідник C# 10. Миттєва допомога для програмістів C# 10. 1-е видання. O'Reilly Media. 2022. С.240 (англ)
14. Joseph Albahari. C# 9.0 in a Nutshell: The Definitive Reference. O'Reilly Media. 2021. С.1060 (англ)
15. Jeffrey Richter. CLR via C# (Developer Reference). Microsoft Press. С.894 (англ)
16. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Модульні курси з дисципліни для дистанційної форми навчання (повний комплект матеріалів)
3. Електронна бібліотека університету.