

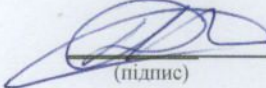
Міністерство освіти і науки України

Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інформаційних технологій проектування (№ 105)

ЗАТВЕРДЖУЮ

Голова НМК 2

 Дмитро КРИЦЬКИЙ
(підпис) (ім'я та прізвище)

«31» 08 2023 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Сучасні технології та інструментарій програмування
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 122 «Комп'ютерні науки»
(код та найменування спеціальності)

Освітня програма: «Інформаційні технології проектування»
(найменування освітньої програми)

Рівень вищої освіти: другий (магістерський)

Харків 2023 рік

Розробник: доцент, к.т.н. каф.105 Сергій КОБА
(ім'я та прізвище, посада, науковий ступінь і вчене звання)



(підпис)

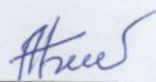
Робочу програму розглянуто на засіданні кафедри 105 інформаційних технологій проектування

(назва кафедри)

Протокол № 1 від «30» серпня 2023 р.

В.о. зав. кафедри 105

_____ (науковий ступінь та вчене звання)



(підпис)

Андрій БИКОВ

(ім'я та прізвище)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)	
Кількість кредитів – 4,5 (2 для КП)	Галузь знань 12 «Інформаційні технології» <small>(шифр та найменування)</small> Спеціальність 122 «Комп'ютерні науки» <small>(код та найменування)</small> Освітня програма «Інформаційні технології проектування» <small>(найменування)</small> Рівень вищої освіти: другий (магістерський)	<i>Обов'язкова</i>	
Кількість модулів – 2		Навчальний рік	
Кількість змістових модулів – 2		2023/2024	
Курсовий проект – «Розробка програмного продукту для вирішення задач, що виникають під час комп'ютеризації інженерної діяльності»		Семестр	
Загальна кількість годин – 64/135		1-й	2-й
Кількість тижневих годин для денної форми навчання: аудиторних – 3,8 (0,9) самостійної роботи студента – 4,7 (3,1)		Лекції*	
		32 годин	
		Практичні, семінарські*	
			16 годин
		Лабораторні*	
	32 годин		
	Самостійна робота		
	71 годин	44 годин	
Вид контролю			
модульний контроль, іспит	диференційований залік		

Співвідношення кількості годин аудиторних занять до самостійної роботи становить для денної форми $64/71 = 0,9$

¹⁾ Курсовий проект та практичні заняття виконуються у другому семестрі на першому році підготовки

²⁾ Аудиторне навантаження може бути зменшеним або збільшеним на одну годину в залежності від розкладу занять.

2. Мета та завдання навчальної дисципліни

Об'єкт вивчення – розподілені застосування Windows, які будуються на засадах моделі компонентних об'єктів (COM) та технології .NET Framework.

Мета: оволодіння технологією побудови взаємодіючих застосувань Windows та використання можливостей складного програмного забезпечення з використанням сучасних технологій, що використовують COM, DCOM, COM+ та .NET Framework.

Предмет дисципліни становить інтегроване середовище розробки програм Microsoft Visual C++, C++/CLI, C#, бібліотека класів Active Template Library (ATL), модель компонентних об'єктів (COM), серверні та клієнтські застосування Windows, платформа .NET Framework взаємодія розподілених програмних продуктів, тестування і налагодження програм, класи та їх використання, механізми інкапсуляції, успадкування, розробка поліморфних об'єктів, розв'язання типових задач опрацювання даних з широким використанням можливостей об'єктно-орієнтованого програмування (ООП), COM, DCOM та COM+.

В результаті вивчення навчальної дисципліни студент повинен:

знати:

- сучасні технології побудови взаємодіючих розподілених програмних продуктів;
- моделі компонентних об'єктів COM, COM+, DCOM та новітньої технології .NET Framework;
- засади програмування застосунків Windows з використанням інтерфейсу API Win32;
- використання застосунків Windows як серверів та контролерів автоматизації;

вміти:

- розробляти алгоритми та програми для вирішення прикладних задач на сучасному рівні з використанням сучасних технологій (COM, COM+, DCOM та .NET Framework);
- проектувати інтерфейс та прикладну частину програмних продуктів;
- розробляти програми для вирішення прикладних задач з широким використанням можливостей сучасних технологій розробки програмного забезпечення;

мати уяву:

- про сучасні тенденції у розвитку технології програмування;
- про загальні проблеми розробки складного програмного забезпечення.

Згідно з вимогами освітньо-наукової програми студенти повинні досягти таких **компетентностей:**

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК05. Здатність вчитися й оволодівати сучасними знаннями.

ЗК07. Здатність генерувати нові ідеї (креативність).

СК01. Усвідомлення теоретичних засад комп'ютерних наук.

СК02. Здатність формалізувати предметну область певного проєкту у вигляді відповідної інформаційної моделі.

СК05. Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.

СК06. Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук.

СК07. Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.

СК08. Здатність розробляти і реалізовувати проекти зі створення програмного забезпечення, у тому числі в непередбачуваних умовах, за нечітких вимог та необхідності застосовувати нові стратегічні підходи, використовувати програмні інструменти для організації командної роботи над проектом.

СК10. Здатність оцінювати та забезпечувати якість ІТ-проектів, інформаційних та комп'ютерних систем різного призначення, застосовувати міжнародні стандарти оцінки якості програмного забезпечення інформаційних та комп'ютерних систем, моделі оцінки зрілості процесів розробки інформаційних та комп'ютерних систем.

СК11. Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.

Очікувані результати навчання:

РН6. Розробляти концептуальну модель інформаційної або комп'ютерної системи.

РН10. Проектувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення

РН11. Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування

РН14. Тестувати програмне забезпечення.

РН17. Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.

РН18. Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується

Пререквізити – «Сучасні технології та інструментарій програмування».

Кореквізити – «Основи ІТ консалтингу».

3. Програма навчальної дисципліни

Модуль 1. Технологія COM та її використання.

Змістовий модуль 1. Вступ до технології COM

Тема 1. Концепції програмування. Огляд особливостей основних концепцій програмування: процедурне програмування, модульне програмування, об'єктно-орієнтоване програмування, модель компонентних об'єктів, загальномовне середовище виконання програм (CLR – Common Language Runtime). Переваги і недоліки перелічених концепцій.

Тема 2. Передумови виникнення концепції компонентних об'єктів. Огляд способів взаємодії застосувань Windows з метою обміну даними. Стислий огляд загальних властивостей технологій DLL, ODBC, DDE і OLE. Взаємодії процесів з використанням буфера обміну (Clipboard), відображення файлів у пам'ять (File Mapping), поштових скриньок (Mailslots), каналів (Pipes), віддаленого виклику процедур (RPC), об'єктів Windows Sockets. Переваги і недоліки перелічених технологій і розробка моделі компонентних об'єктів COM.

Тема 3. Тип Interface як ключове поняття технології COM. Поняття типу Interface як ключового в технології COM. Глобально-унікальні ідентифікатори. Базовий інтерфейс IUnknown. Показники на інтерфейси COM об'єктів. Найпростіші приклади використання типу інтерфейс у програмах на мовах Object Pascal і C++.

Тема 4. Загальні засади технології COM. Технологія COM як специфікація і реалізація. Вимоги, яким повинні відповідати застосування, що використовують технологію COM. Розширення COM. Реалізація COM у мовах Object Pascal і C++. Класи, призначені для підтримки використання технології COM, і відповідні інструментальні засоби.

Тема 5. Структурні та ключові елементи технології COM. Сервери COM і клієнти. Клас CoClass і фабрика класів. Типи серверів та їх реалізація. Сервер у клієнтському процесі (in-process server). Локальний сервер (local server). Віддалений сервер (remote server). Переваги і недоліки перелічених серверів. Механізм маршалінгу.

Змістовий модуль 2. Технологія компонентних об'єктів

Тема 6. Розробка власних об'єктів COM та відповідних клієнтських застосувань. Алгоритм створення найпростішого COM об'єкта. Проектування COM об'єкта. Майстер створення COM об'єкта. Вибір способу створення екземпляра об'єкта (instanting mode). Модель потокової обробки (threading model). Реєстрація і тестування COM об'єкта. Опис і ієрархія сокласу (coclass), призначення його конструкторів. Послідовність виконання функцій при одержанні клієнтським застосуванням показника на інтерфейс. Методи, експортовані бібліотекою сервера.

Тема 7. Бібліотеки типів. Поняття бібліотеки типів та переваги її використання. Необхідність імпорту бібліотеки типів. Способи імпорту бібліотеки типів та її використання.

Тема 8. Інтерфейси COM. Види інтерфейсів: базовий інтерфейс IUnknown, диспетчерський інтерфейс IDispatch, дисінтерфейс (dispinterface) і двоїтий інтерфейс (dual interface). Методи інтерфейсів та їх призначення. Особливості опису і використання інтерфейсів.

Тема 9. Створення клієнтських застосувань COM. Способи створення клієнтських застосувань. Генерація інтерфейсного модуля для сервера шляхом імпорту бібліотеки типів. Використання необхідного застосування як сервера автоматизації.

Модуль 2. Технологія .NET Framework та її використання.

Змістовий модуль 3. Вступ до технології .NET Framework

Тема 10. Склад і призначення .NET Framework. Передумови виникнення .NET Framework як сучасної концепції розробки застосувань Windows. CLR та FCL як складові технології .NET Framework. Простір імен System та його складові частини.

Тема 11. Процес розробки додатків на платформі CLR. Збірка (assembly) NET Framework та її основні частини. Що є just-in-time (JIT) компілятор, що подається йому на вхід і що виходить на виході. Метадані збірки. Принципові відмінності виконання додатка на платформі CLR від виконання «старих» додатків Windows. Типи файлів, у яких може зберігатися збірка (assembly) NET Framework

Тема 12. Основи типів. Сумісність типів. Вживання операцій is та as. Правила вживання операції as. Призначення операції is. Відмінності операцій.

Змістовий модуль 4. Використання технології .NET Framework

Тема 13. Елементарні, посилальні і значимі типи. Поняття типів. Переваги та недоліки посилальних і значимих типів. Призначення та використання операторів checked и unchecked. Конструктори типів та екземплярів. Відмінності між статичними і екземплярами методами класів.

Тема 14. Методи класів: конструктори, оператори, перетворення і параметри. Константи і поля. Призначення та формат опису і використання властивостей. Ключове слово Initonly і в чому його сенс, призначення. Приклади опису властивостей класу. Поняття «властивості з параметрами», його опис та використання.

Тема 15. Інтерфейси (interface) та делегати (delegate). Призначення та опис інтерфейсів як мовних засобів. Призначення та опис узагальнених інтерфейсів. Призначення та опис делегатів як мовних засобів.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	денна форма				
	усього	у тому числі			
л		п	лаб	курс	с.р.
Модуль 1. Технологія COM та її використання					
Змістовий модуль 1. Вступ до технології COM					
Тема 1. Концепції програмування.	1	1	-	-	-
Тема 2. Передумови виникнення концепції компонентних об'єктів.	10	1	-	-	4
Тема 3. Тип Interface як ключове поняття технології COM.	7	1	-	-	6

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	курс	с.р.	
Тема 4. Загальні засади технології COM.	10	2	-	2		6
Тема 5. Структурні та ключові елементи технології COM.	13	1	-	4		8
Разом за змістовим модулем 1	36	6	-	6		24
Змістовий модуль 2. Технологія компонентних об'єктів						
Тема 6. Розробка власних об'єктів COM та відповідних клієнтських застосувань.	12	4	-	4		4
Тема 7. Бібліотеки типів.	8	2	-	-		6
Тема 8. Інтерфейси COM.	8	2	-	-		6
Тема 9. Створення клієнтських застосувань COM.	18	2	-	6		10
Разом за змістовим модулем 2	46	10	-	10		26
Усього годин за модулем 1	82	16	-	16		50
Модуль 2. Технологія .NET Framework та її використання						
Змістовий модуль 3. Вступ до технології .NET Framework						
Тема 10. Склад і призначення .NET Framework.	10	2	-	-		8
Тема 11. Процес розробки додатків на платформі CLR.	19	3	-	4		12
Тема 12. Основи типів	17	3	-	4		10
Разом за змістовим модулем 3	46	8	-	8		30
Змістовий модуль 4. Використання технології .NET Framework						
Тема 13. Елементарні, посилальні і значимі типи	15	2	-	-		13
Тема 14. Методи класів: конструктори, оператори, перетворення і параметри.	18	2	-	4		12
Тема 15. Інтерфейси (interface) та делегати (delegate).	18	4	-	4		10
Разом за змістовим модулем 4	51	8	-	8		35
Усього годин за модулем 2	97	16	-	16		65
Курсовий проект	60	-	16	-		44
Усього годин за дисципліною	195	32	16	32		115

5. Теми семінарських занять

Семінарські заняття навчальним планом не передбачені.

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Опис та використання узагальнених класів	2
2	Створення та використання серверу, що розміщений у адресному просторі клієнтського застосування	2
3	Створення та використання локального сервера	3
4	Розробка класів з використанням інтерфейсів	3
5	Використання делегатів для обробки подій (event)	2
6	Оброблення виняткових ситуацій	4
	Разом	16

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Розроблення типового консольного додатка динамічно приєднувана бібліотека	2
2	Додаток, що використовує динамічно приєднувану бібліотеку	2
3	Розробка та використання DLL-розширення та його клієнта	2
4	Розроблення і використання бібліотек DLL	2
5	Розроблення серверу та клієнтського додатку з використанням ATL	4
6	Розроблення і використання локального сервера	4
7	Сервер графіки	4
8	Розроблення контейнера ActiveX	2
9	Розроблення і використання елементів керування ActiveX	2
10	Розробка додатків . NET для платформи CLR	4
11	Доступ до серверу Excel на мові C/CLI	4
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Функції, що призначені для керування процесами	5
2	Створення та керування потоками	6
3	Завантаження бібліотеки DLL та виклик її функцій	6
4	Еволюція розподілених систем. Проблеми взаємодії застосувань Windows: походження та способи їх вирішення.	6
5	Функції базового інтерфейсу. Генерування унікальних ідентифікаторів GUID. Призначення та способи використання MTS та MSMQ	6
6	Технологія COM як специфікація та реалізація. Вимоги COM до мов програмування, що її використовують	6
7	Мова IDL, що призначена для опису інтерфейсів. Відмінності опису інтерфейсів у різних мовах програмування. Застосування, що призначені для розроблення опису інтерфейсів	6

8	Функції Win API, що призначені для взаємодії клієнтських застосунків та серверів. Послідовність отримання інтерфейсів та функцій серверів	6
9	Призначення та склад бібліотеки ATL, її недоліки та переваги порівняно з іншими бібліотеками подібного призначення	6
10	Передумови, що призвели до необхідності створення технології OLE. Технології, що передували OLE: приклади застосування та їх недоліки.	6
11	Особливості використання алгоритмів стандартної бібліотеки у застосуваннях, що використовують технологію COM	6
12	Інтерфейси та делегати	6
13	Курсовий проект	44
	Разом	115

9. Індивідуальне завдання (курсний проект)

Мета виконання – набуття практичних навичок самостійного розроблення сучасних програмних засобів, призначених для вирішення задач, що виникають під час комп'ютеризації інженерної діяльності. У процесі виконання курсового проекту студенти також закріплюють ті практичні знання та навички, набуті при вивченні дисципліни.

Індивідуальні теми курсових робіт надаються студентам у окремому електронному документі і студенти мають можливість самостійно обрати тему проекту із запропонованих або запропонувати свою тему, що відповідає вимогам до проекту. Технічне завдання (ТЗ) розробляється студентом у першу чергу згідно з наведеним нижче графіком виконання курсового проекту.

Склад пояснювальної записки повинен бути таким:

1. Титульний аркуш
2. Зміст
3. Вступ
4. Технічне завдання
5. Проектування додатка
6. Керівництво користувача
7. Керівництво програміста
8. Висновок
9. Перелік посилань

Обсяг записки не обмежується і, в середньому, сягає 20-30 сторінок.

Структура ТЗ наступна:

1. Призначення програмного продукту, що включає повний перелік його функцій.
2. Опис вхідних даних із зазначенням джерела (дисковий файл, введення в діалоговому режимі, таблиці БД і т.п.).
3. Опис вихідних даних – результатів.
4. Характер і структура інтерфейсу користувача: ескізи екранних форм, опис меню програми та інструментальних панелей, передбачуваних до розробки.
5. Технічні та програмні ресурси, необхідні для розробки і експлуатації програмного продукту.

ТЗ розробляється в першу чергу, підписується автором і викладачем і згодом включається в пояснювальну записку в якості її складової частини.

Проектування додатка. На цьому етапі повинні бути виконані перераховані нижче роботи та їх результати повинні бути відображені у відповідних розділах пояснювальної записки.

Аналіз предметної області (місце докладання в ряду аналогічних програмних продуктів, необхідність і актуальність його розробки і т.д.) і наявних прототипів. Вхідні дані та результати роботи програми із зазначенням їх розмірностей, загальноприйнятих позначень. Функціональний і структурний аналіз розв'язуваної задачі. Математична модель розв'язуваної задачі (якщо така є) і виклад алгоритму.

Інструкція користувача. Середовище виконання додатку, тобто версія(ї) операційної системи. Перелік файлів, необхідних для нормального функціонування програми, із зазначенням їх призначення. Якщо для роботи програми потрібна наявність в системі додаткових програмних продуктів, то їх також необхідно вказати. Тут треба вказати саме ті файли, які необхідні для запуску готового додатку на виконання, тобто файли з вихідними текстами програм вказувати не треба. Якщо файли програми повинні розміщуватися в певних каталогах, то їх структуру також необхідно надати. Детальний опис інтерфейсу додатку, послідовність дій користувача. Використання системи допомоги. Обмеження, можливі помилки і дії користувача при їх виникненні. Приклади використання програми.

Керівництво програміста. Повний перелік (вихідних) файлів програми із зазначенням їх призначення. Без гострої необхідності в перелік цих файлів не варто включати відкомпільовані файли модулів та інші допоміжні файли на кшталт .Obj, .Pch, .Sbr і т.п. Структура програми на рівні компонентів і класів. Ієрархія (власних) класів, що використовуються в додатку. Опис класів, що включає опис їхніх атрибутів, властивостей і методів. Опис класу повинен, в обов'язковому порядку, починатися з оголошення тіла класу, тобто вмісту відповідного заголовного файлу. Також повинні бути зроблені описи глобальних змінних, тобто описи, зроблені поза підпрограм і класів.

Висновок. Висновки про відповідність розробленого додатку завданню.

Термін виконання курсового проекту – тижні 1 – 14. Трудомісткість – 16 годин аудиторних занять, 44 година самостійної роботи. Типовий план-графік виконання курсового проекту такий:

№	Етап	Термін (Тиждень семестру)	Вид звіту	% готовності
1	Розробка і затвердження технічного завдання	3	Розділ записки (тверда копія)	20
2	Проектування додатка	5	Розділ записки (електронна форма)	35
4	Кодування додатки - перша працююча версія	10	Програма	50
5	Налагодження і тестування програми	15	- -	80
6	Оформлення записки	16	Записка	90
7	Захист проекту	17	Записка і програма	100

Варіанти курсових проектів надані на сервері кафедри. Для магістрів та студентів, залучених до виконання науково-дослідних робіт, можливе виконання курсових проектів дослідницького спрямування, узгоджених з індивідуальними планами підготовки.

10. Методи навчання

Лекції проводяться з використанням основних розділів конспекту лекцій в електронній формі, елементів мультимедійної підтримки курсу (відео фрагментів), демонстрацій окремих прийомів роботи з інструментальним середовищем та/або роздаткових матеріалів у вигляді схем та діаграм.

Лабораторні роботи виконуються з використанням навчальних (демонстраційних) та ліцензованих робочих версій сучасного програмного забезпечення.

Самостійна робота включає підготовку до лабораторних робіт, модульного контролю та іспиту, виконання поза-аудиторної частини індивідуального завдання і вивчення вказаних вище тем за конспектом, літературними джерелами та програмною документацією.

11. Методи контролю

Контроль здійснюється згідно з «Положенням про модульно-рейтингову систему оцінювання знань студентів».

Поточний контроль – відповідно до повноти, якості та своєчасності виконання лабораторних робіт, індивідуального завдання; проміжний (модульний) контроль – письмові контрольні роботи на 8-му та 16-му тижнях; підсумковий контроль – письмовий іспит.

12. Критерії оцінювання та розподіл балів, які отримують студенти

Розподіл балів, які отримують студенти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	0...2	5	0...10
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	10...15	1	10...25
Змістовний модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	10...15	1	10...25
Усього за семестр			60...100

Семестровий контроль (іспит/залік) проводиться у разі відмови студента від балів поточного тестування та за наявності допуску до іспиту/заліку. При складанні семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з двох теоретичних питань та одного практичного завдання. Максимальна кількість балів за кожне питання - 30 балів, за практичне завдання – 40 балів, загальна сума – 100 балів.

Критерії оцінювання роботи здобувача протягом семестру

Необхідний обсяг знань для одержання позитивної оцінки:

- мати уявлення про весь теоретичний матеріал дисципліни;
- знати в повному обсязі не менш половини тем теоретичного матеріалу;
- знати додатковий матеріал по трьом темам.

Необхідний обсяг умінь для одержання позитивної оцінки:

- володіти технологією застосування теоретичних знань на практиці;
- уміти самостійно знаходити довідкові відомості;
- уміти застосовувати на практиці отримані знання.

Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Знати основні положення теоретичного матеріалу. Вміти користуватися усіма засобами інструментального середовища Visual Studio. Вміти розробляти додатки та динамічні бібліотеки Windows, що вирішують задачі з використанням функцій та класів.

Добре (75-89). Знати основний теоретичний матеріал в повному обсязі. Володіти технологією пошуку довідкової літератури. Вміти розробляти та налагоджувати додатки в інструментальному середовищі Visual Studio, що реалізують технологію COM.

Відмінно (90-100). Знати основний і додатковий теоретичний матеріал в повному обсязі. Добре орієнтуватися в довідковій літературі. Вміти вирішувати задачі та розробляти додатки з використанням віртуальних функцій, шаблонних класів, технологій COM та OLE.

Переведення підсумкових семестрових оцінок в оцінки за національною шкалою та в оцінки за міжнародною шкалою ECTS здійснюється відповідно до «Положення про модульно-рейтингову систему оцінювання знань студентів» за наведеною нижче таблицею.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90–100	A	відмінно
83–89	B	добре
75–82	C	
68–74	D	задовільно
60–67	E	
0–59	FX	незадовільно

13. Методичне забезпечення

Завдання та сценарії виконання лабораторних робіт у електронній формі, допоміжні приклади додатків Windows, що надаються студентам на сервері кафедри.

14. Рекомендована література

14.1. Базова

1. Кривцова О.П. К82 Програмування мовою C++. Технологія візуального програмування: навч. посіб. – Полтава : ПНПУ імені В.Г. Короленка, 2020. – 144 с.

2. Основи програмування на C++ [Електронний ресурс] : навч. посібник / О. О. Водка [та ін.] ; Нац. техн. ун-т "Харків. політехн. ін-т". – Електрон. текст. дані. – Харків, 2021. – 112 с. – URI: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/52280>.

3. Трофименко О. Г. С++. Алгоритмізація та програмування: підручник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, О. В. Задерейко. 2-ге вид. перероб. і доповн. – Одеса: Фенікс, 2019. – 477 с.

4. Фрімен Ерік. Head First. Патерни проєктування / Ерік Фрімен, Елізабет Робсон, Кеті Сьєрра і Берт Бейтс; пер. з англ. Г. Якубовська — Харків : ВД «Фабула», 2020. — 672 с.

5. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фабула», 2019. – 448 с.

14.2. Допоміжна

1. Price, Mark J. *C# 8.0 and .NET Core 3.0–Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code*. Packt Publishing Ltd, 2019.

2. Albahari, Joseph, and Ben Albahari. *C# 9.0 Pocket Reference*. " O'Reilly Media, Inc.", 2021.

3. Sharp, John. *Microsoft Visual C# Step by Step*. Microsoft Press, 2018.