

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ТРАНСПОРТНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
П.2.1. КРОСПЛАТФОРМНІ ТЕХНОЛОГІЇ В РОЗРОБЦІ
ІНФОРМАЦІЙНИХ СИСТЕМ

рівень вищої освіти	магістр <i>перший (бакалаврський) / другий (магістерський)</i>
галузь знань	12 «Інформаційні технології» <i>шифр і назва</i>
спеціальність	121 Інженерія програмного забезпечення <i>шифр і назва</i>
освітня програма / спеціалізація	«Інженерія програмного забезпечення» <i>повна назва</i>
тип дисципліни	обов'язкова <i>обов'язкова/вибіркова/факультативна</i>
мова(и) викладання	українська <i>українська / англійська/німецька/російська</i>

2024-2025 н.р.

Робоча програма навчальної дисципліни «Кросплатформні технології в розробці інформаційних систем» для підготовки фахівців галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення», що навчаються за освітньою програмою «Інженерія програмного забезпечення» для здобуття другого (магістерського) рівня освіти.

РОЗРОБНИКИ ПРОГРАМИ:

Сватко В.В., доцент, к.т.н., доцент

Робочу програму схвалено на засіданні кафедри інформаційних систем і технологій

Протокол № 1 від «26» серпня 2024 року

ПОГОДЖЕНО на засіданні Вченої ради факультету транспортних та інформаційних технологій

Протокол № 1 від «27» серпня 2024 року

1.Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, рівень вищої освіти, ступінь	Характеристика навчальної дисципліни	
		денна форма навчання	
Кількість кредитів – 5,00	Галузь знань 12 «Інформаційні технології»	нормативна	
Модулів – 2	Спеціальність: 121 «Інженерія програмного забезпечення»	Рік підготовки	
Індивідуальне комплексне завдання –		1-й	
Загальна кількість годин - 150		Семестр	
		1-й	
		Лекції	
Тижневих годин для денної форми навчання: аудиторних –2 самостійної роботи студента -3,63	Другий (магістерський) рівень вищої освіти: ступінь – магістра	15 годин	
		Практичні	
		-	
		Лабораторні	
		30 годин	
		Самостійна робота	
		105 години	
		Індивідуальні завдання:	
		38 годин	
		Вид контролю:	
екзамен			

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної роботи і індивідуальної роботи становить (%):

Для денної форми навчання – 40%

2. Мета та завдання навчальної дисципліни

Мета - теоретичне вивчення та отримання практичних навичок у питаннях:

- принципів технології розробки крос-платформних програмних систем;
- принципів використання засобів крос-платформного програмування.

Міждисциплінарні зв'язки дисципліни «Крос-платформні технології в розробці інформаційних систем» наведено на рисунку 1.



Рисунок 1- Схема міждисциплінарних зв'язків дисципліни «Крос-платформні технології в розробці інформаційних систем»

Основним завданням вивчення дисципліни «Крос-платформні технології в розробці інформаційних систем» є формування у студентів знань і навичок з методології та технології розробки крос-платформних програмних систем; принципів використання засобів крос-платформного програмування.

У результаті вивчення навчальної дисципліни студент повинен

знати :

- базові принципи створення крос-платформних програмних систем;
- теоретичні знання та практичні навички у галузі застосування засобів крос-платформного програмування;
- сучасні методи та технології створення платформи-незалежного програмне забезпечення;

вміти :

- проектувати платформи-незалежні додатки;
- застосовувати на практиці принципи компонентно-орієнтованого проектування;
- створювати крос-платформні додатки з використанням особливостей компонентних технологій.

В освітній програмі підготовки бакалавра з комп'ютерних наук студенти в результаті вивчення дисципліни «Крос-платформні технології в розробці інформаційних систем» набувають такі компетентності:

Загальні компетентності (за вимогами НРК):

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК03. Здатність проводити дослідження на відповідному рівні.

ЗК04. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами інших галузей знань/видів економічної діяльності).

Фахові компетентності спеціальності:

СК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.

СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.

СК03. Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.

СК04. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.

СК05. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.

СК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.

СК09. Здатність забезпечувати якість програмного забезпечення.

СК10. Здатність до аналізу і застосування сучасних інформаційних технологій при моделюванні транспортних процесів.

Програмні результати навчання.

РН02. Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

РН03. Будувати і досліджувати моделі інформаційних процесів у прикладній області.

РН06. Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

РН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.

PH09. Обґрунтовано вибрати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

PH10. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.

PH13. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

PH16. Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

PH18 Оцінювати і вибирати ефективні методи сучасних інформаційних технологій при моделюванні транспортних процесів.

3. Програма навчальної дисципліни

Модуль 1. Принципи крос-платформного програмування

1. Крос-платформні технології

Основні архітектури програмного забезпечення. Стан корпоративної програмного середовища типовою організації.

2. Компонентна ідеологія

Визначення та властивості компонентів. Специфікація інтерфейсу як контракту.

3. Стратегії інтеграції програмного забезпечення

Модель посилань. Стратегії інтеграції програмного забезпечення.

4. Методи створення компонентів

Розробка та збирання компонентів. Об'єкти та сервіси, що ними надаються.

Модуль 2. Розробка крос-платформних програм. Компонентні технології

5. Архітектура та проектування компонентних систем

Розподілена архітектура компонентних систем. Компонентно-орієнтоване проектування. Формальні та візуальні методи конструювання компонентів.

6. Проміжне програмне забезпечення

Брокери об'єктних запитів. Монітори оброблення транзакцій. Виклики розподілених методи. Вибір додатків, сервісів, компонентів і протоколів зв'язку.

7. Особливості компонентних технологій

COM/DCOM/NET. CORBA. Java Beans.

8. Web-технології.

Основні технології Web-програмування. Аплети. Сервлети.

4. Структура навчальної дисципліни

При вивченні даної дисципліни студент має ознайомитися з програмою дисципліни, з її структурою, формами та методами навчання, видами та методами контролю знань.

Тематичний план навчальної дисципліни складається з двох модулів, що логічно пов'язують кілька навчальних елементів дисципліни за змістом та взаємозв'язками.

Навчальний процес здійснюється у таких формах: лекційні, лабораторні заняття та самостійна робота студента.

Структура залікового кредиту з дисципліни наведено у таблиці нижче.

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	Усього	у тому числі				
Лекції		Прак-тичні	Лабора-торні	Індив. завдання	Самост. робота	
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
Модуль 1. Принципи крос-платформного програмування						
1. Крос-платформні технології						
Тема 1. Основні архітектури програмного забезпечення.	8	1		1	2	4
Тема 2. Стан корпоративної програмного середовища типовою організації	9	1		2	2	4
Разом	17	2		3	4	8
2. Компонентна ідеологія						
Тема 1. Визначення та властивості компонентів.	9	1		2	2	4
Тема 2. Специфікація інтерфейсу як контракту.	9	1		2	2	4
Разом	18	2		4	4	8
3. Стратегії інтеграції програмного забезпечення						
Тема 1. Модель посилань	9	1		2	2	4
Тема 2. Стратегії інтеграції програмного забезпечення.	9	1		2	2	4
Разом	18	2		4	4	8
4. Методи створення компонентів						
Тема 1. Розробка та збирання компонентів. Об'єкти та сервіси, що ними надаються.	22	1		4	4	13
Разом	22	1		4	4	13
Разом за модулем 1	75	7		15	16	37

Модуль 2. Розробка крос-платформних програм. Компонентні технології						
5. Архітектура та проектування компонентних систем						
Тема 1. Розподілена архітектура компонентних систем. Компонентно-орієнтоване проектування. Формальні та візуальні методи конструювання компонентів.	18	2		4	4	8
Разом	18	2		4	4	8
6. Проміжне програмне забезпечення						
Тема 1. Брокери об'єктних запитів. Монітори оброблення транзакцій.	9	1		2	2	4
Тема 2. Виклики розподілених методів. Вибір додатків, сервісів, компонентів і протоколів зв'язку	9	1		2	2	4
Разом	18	2		4	4	8
7. Особливості компонентних технологій						
Тема 1. COM/DCOM/NET. CORBA. Java Beans.	17	2		3	4	8
Разом	17	2		3	4	8
8. Web-технології						
Тема 1. Основні технології Web-програмування. Аплети. Сервлети.	22	2		4	10	6
Разом	22	2		4	10	6
Разом за модулем 2	75	8		15	22	30
ІНДЗ					38	
Усього годин	150	15		30	38	67

5. Теми семінарських занять

Семінарські заняття навчальним планом дисципліни «Крос-платформні технології в розробці інформаційних систем» не передбачені.

6. Теми практичних занять

Практичні заняття навчальним планом дисципліни «Крос-платформні технології в розробці інформаційних систем» не передбачені.

7. Теми лабораторних занять

На кожному лабораторному занятті до виконання лабораторної роботи студент має відповісти на контрольні питання, які відображають його готовність до виконання лабораторної роботи, зокрема оволодіння необхідними теоретичними знаннями та усвідомлення мети роботи. По закінченні виконання лабораторної роботи викладач оцінює ступінь оволодіння навичками та досягнення мети даної роботи.

Для задачі лабораторної роботи студенту необхідно оформити індивідуальний звіт, у якому повинна бути: постановка завдання, роздруковані основні результати роботи, аналіз розрахунків та чіткі висновки.

Підсумкові оцінки за виконання кожної лабораторної роботи вносяться у відповідний журнал. Отримані студентом оцінки за лабораторні роботи враховуються при виставленні підсумкової оцінки з даної навчальної дисципліни.

№ з/п	Назва теми	К-сть годин
1	Бібліотека Juce. Інсталяція	2
2	Початок роботи з Juce. Створення додатків. Головне вікно	4
3	Компонентна модель Juce. Загальні властивості компонентів	4
4	Розробка власних компонентів	4
5	Візуальне проектування компонентів	4
6	Розробка простих платформи-незалежних додатків у JUCE	4
7	Канвас. Малювання та додавання компонентів на канвас	4
8	Використання бібліотеки компонентів для створення крос-платформних додатків	4
Разом за семестр		30

8. Самостійна робота

Розподіл годин самостійної роботи

Всього годин - 72	
ПМК – підготовка до модульного контролю	2 (2 години на семестр)
ПЛ – підготовка до лабораторних занять	32 (до 2 годин на пару)
ІНДЗ - Індивідуальне навчально-дослідне завдання (курсова робота)	38

Розподіл годин самостійної роботи за темами

№ з/п	Назва теми	Кількість Годин	
		ПП	ІКЗ
1.	Тема 1. Використання інструментів Java. Перемикачі	4	4
2.	Тема 2. Об'єкти Меню. Бібліотеки АWT	4	4
3.	Тема 3. Інструментальні кнопки та панелі	4	4
4.	Тема 4. Компоненти ієрархічних структур	4	4
5.	Тема 5. Java-аплети	4	4
6.	Тема 6. Процеси та потоки	4	4
7.	Тема 7. Програмування для комп'ютерних мереж	4	4
8.	Тема 8. Розробка Help-системи. Підключення Help-системи до проекту	4	10
	Всього	32	38
	Підготовка до модульного контролю №1	2	
	Підготовка до підсумкового контролю (заліку)	-	
	Всього	34	38

Розподіл годин за етапами виконання курсової роботи (індивідуального навчально-дослідного завдання)

№	Етапи виконання роботи	Годин	Навчальні тижні															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Отримання завдання до курсової роботи	2	2															
2	Опрацювання літератури та інших інформаційних джерел	6		2	2	2												
3	Складання плану курсової роботи	4				4												
4	Написання основних розділів роботи																	
	4.1. Виконання розрахунків та реалізація практичного завдання	10					2	2	2	2	2							
	4.2. Написання теоретичного розділу	6										2	2	2				
5	Написання вступу та висновку	4													4			
6	Загальне оформлення роботи та здача її на перевірку	2														2		
7	Захист курсової роботи	4															4	
		38	2	2	2	2	4	2	2	2	2	2	2	2	4	2	4	

9. Індивідуальні завдання

Індивідуально-консультативна робота полягає в проведенні консультацій студентів з виконання лабораторних та самостійних завдань, їх перевірки та захисту. Індивідуально-консультативна робота проводиться керівниками курсу, не рідше одного разу на тиждень згідно графіку, що складається на кожен семестр окремо.

Крім вищенаведеного, студентам слід звернути увагу на такі аспекти вивчення науки:

- більш поглиблене опрацювання теоретичних основ прослуханого лекційного матеріалу із використанням наукових ресурсів мережі Internet.

Теми завдань для індивідуальної роботи студентів:

Використовуючи крос-платформну бібліотеку Juice мови C++ створити платформу незалежний додаток згідно з варіантом завдання:

1. Розробити додаток системи продажу жд квитків
2. Розробити додаток системи продажу авіа квитків
3. Розробити додаток системи продажу книг
4. Розробити додаток системи продажу автозапчастин
5. Розробити додаток системи замовлення піци
6. Розробити додаток системи замовлення суші
7. Розробити додаток системи замовлення ліків у аптеці
8. Розробити текстовий редактор
9. Розробити органайзер
10. Розробити калькулятор
11. Розробити додаток для обробки матриць
12. Розробити додаток «Англійсько-український словник»
13. Розробити додаток для перегляду та коментування фотографій
14. Розробити додаток для прослуховування та коментування аудіо записів
15. Розробити гру гра хрестики-нулики
16. Розробити гру «Перший мільйон»
17. Розробити гру «Сапер»
18. Розробити додаток для малювання схеми перехресть
19. Розробити додаток для малювання схеми ДТП
20. Розробити додаток для малювання блок-схем

10. Методи навчання

При вивченні курсу «Крос-платформні технології в розробці інформаційних систем» застосовуються 3 групи методів навчання:

- методи організації і здійснення навчально-пізнавальної діяльності;
- методи стимулювання і мотивації навчально-пізнавальної діяльності;
- методи контролю і самоконтролю за ефективністю навчально-пізнавальної діяльності.

Перша група охоплює вербальні методи передачі і сприймання навчальної інформації (розповідь, лекція); наочні (ілюстрація, презентація); практичні (вправи, групові та індивідуальні завдання). В межах самостійної роботи – робота з книгами, методичними матеріалами, Інтернет-джерелами, творчі завдання.

При вивченні курсу активно використовуються інтерактивні методи (при веденні лекцій та семінарських занять) та проблемно-пошукові методи навчання (як при веденні аудиторних занять, так і при організації самостійної роботи студентів).

11. Методи контролю

Методи поточного контролю: індивідуальне опитування, фронтальне опитування, перевірка лабораторних робіт, перевірка індивідуальних завдань.

Методи модульного контролю: письмова контрольна робота, модульні контролі.

Методи підсумкового контролю: екзамен (2 теоретичних питання та 1 практичне).

Засоби діагностики модульного контролю:

Модульний контроль проводиться у формі тестування. Тест складається з 20 теоретичних питань, для яких необхідно обрати правильні варіанти відповідей із запропонованих.

Приклад тесту, що отримує студент при проходженні модульного контролю

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ			
Кафедра	Інформаційних систем і технологій		
Дисципліна	Крос-платформні технології в розробці інформаційних систем		
ПІБ	Група	Дата	Підпис

ВАРІАНТ №1

1. Juce— відкритий багатоплатформовий інструментарій програмування (фреймворк) для розробки GUI додатків і плагінів мовою
 - C++
 - C#
 - Java
2. Які середовища розробки і компілятори підтримує Juce
 - GCC,Xcode,Visual Studio.
 - Visual Studio.
 - GCC,Visual Studio.
3. Juce не дуже підходить для додатків, інтегрованих в те чи інше графічне середовище
 - оскільки принципово не може емулювати зовнішній вигляд системних компонентів / віджетів,
 - оскільки не використовує додаткові бібліотеки
4. Допоміжна програма, яка входять до складу Juce і відповідає за візуальне проектування компонентів:
 - the jucer
 - Introjucer
5. Допоміжна програма, яка входять до складу Juce і відповідає за створення проектів:
 - the jucer
 - Introjucer
6. Основою віконного додатка Juce є клас, що успадковує
 - DocumentWindow.
 - START_JUCE_APPLICATION
 - JUCEApplication
7. Із запропонованих варіантів оберіть макрос, який здійснює запуск програми, написаної з використанням Juce:
 - JUCEApplication
 - TMainForm
 - START_JUCE_APPLICATION
8. Оберіть із запропонованих варіантів клас, який зберігає інформацію про колір:
 - Colour
 - Coloring
 - ColourFond
9. Оберіть із запропонованих варіантів клас, який відповідальний за створення вікна із змінними розмірами і смугою заголовка:
 - ResizableWindow
 - DocumentWindow
 - JUCEApplication

10. Оберіть із запропонованих варіантів базовий клас для переміщуваних вікон:

- ResizableWindow
- DocumentWindow
- JUCEApplication

11. Оберіть із запропонованих варіантів метод, який повертає рядок - назва програми:

- getApplicationTitle()
- getApplicationMain()
- getApplicationName()

12. Оберіть із запропонованих варіантів метод, який повертає рядок, що містить номер версії додатка:

- getApplicationVersion()
- getApplicationNumber()
- getApplicationVersionNumber()

13. Технологія CORBA –

- визначає, яким чином програмні компоненти, розподілені по мережі, можуть взаємодіяти один з одним незалежно від оточуючих їх операційних систем та мов реалізації;
- це стандарт Microsoft, що визначає структуру та взаємодію компонентів програмного забезпечення в сучасних операційних системах MS Windows;
- це компонентна архітектура, призначена для розробки і розгортання розподілених бізнес-додатків, заснованих на компонентах.

14. Технологія Enterprise Java Beans –

- визначає, яким чином програмні компоненти, розподілені по мережі, можуть взаємодіяти один з одним незалежно від оточуючих їх операційних систем та мов реалізації;
- це стандарт Microsoft, що визначає структуру та взаємодію компонентів програмного забезпечення в сучасних операційних системах MS Windows;
- це компонентна архітектура, призначена для розробки і розгортання розподілених бізнес-додатків, заснованих на компонентах.

15. Технологія JINI -

- визначає, яким чином програмні компоненти, розподілені по мережі, можуть взаємодіяти один з одним незалежно від оточуючих їх операційних систем та мов реалізації;
- являє собою технологію створення розподілених систем, орієнтовану виключно на використання Java;
- це стандарт Microsoft, що визначає структуру та взаємодію компонентів програмного забезпечення в сучасних операційних системах MS Windows.

16. Технологія SOAP –

- визначає, яким чином програмні компоненти, розподілені по мережі, можуть взаємодіяти один з одним незалежно від оточуючих їх операційних систем та мов реалізації;
- являє собою технологію створення розподілених систем, орієнтовану виключно на використання Java;

- полягає в обміні повідомленнями між віддаленими об'єктами по протоколу HTTP з використанням XML в якості транспорту.

17. Сервлети це-

- програми на Java, які працюють на серверному комп'ютері;
- програми на Java, що працюють під управлінням іншої програми та програми на Java, які працюють на серверному комп'ютері;
- програми на Java, що працюють під управлінням іншої програми.

18. Аплети – це

- програми на Java, які працюють на серверному комп'ютері;
- програми на Java, що працюють під управлінням іншої програми та програми на Java, які працюють на серверному комп'ютері;
- програми на Java, що працюють під управлінням іншої програми.

19. До дволанкової архітектури програмного забезпечення "клієнт-сервер" належать:

- сервісні програми, системні утиліти, текстові та графічні редактори, компілятори, досить прості корпоративні програми;
- Web-технології (клієнт-браузер, сервер-Web-сервер), робота з розподіленими СУБД (клієнт - спеціальна програма, сервер - сервер бази даних);
- великі корпоративні системи, які мають на меті охопити весь спектр діяльності підприємства.

20. До основних технологій Web-програмування не належать:

- CGI-скрипти;
- Спеціальні інтерпретовані мови скриптів;
- GUI-скрипти;
- Аплети.

Засоби діагностики підсумкового контролю (екзамен)

Екзамен здійснюється письмово та за комп'ютером (індивідуальне практичне завдання) за екзаменаційними білетами. Екзаменаційний білет складається з трьох питань. Результат екзамену оцінюються за 40-бальною системою. Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання.

Приклад екзаменаційного білета з дисципліни:

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ

Рівень вищої освіти	другий (магістерський)
Галузь знань	12 – Інформаційні технології
Спеціальність	121 – Інженерія програмного забезпечення
Семестр	1
Навчальна дисципліна	Крос-платформні технології в розробці інформаційних систем

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

1. Основні апаратні та програмні платформи
2. Архітектура процесора RISC
3. Індивідуальне практичне завдання

Затверджено на засіданні
кафедри інформаційних систем і технологій
Протокол №1 від 28 серпня 2023 року

Завідувач кафедри ІСТ _____ Валерій ГАВРИЛЕНКО

Екзаменатор _____ Віталій СВАТКО

12. Розподіл балів, які отримують студенти

Модулі	Модуль I										Модуль II										Кількість балів у семестрі*	Підсумковий контроль (залік, екзамен)**						
Кількість балів за модуль	30										Модульний контроль	30											Модульний контроль					
Змістові модулі	ЗМ 1			ЗМ 2			ЗМ 3			ЗМ 4				ЗМ 5		ЗМ 6		ЗМ 7		ЗМ 8								
Кількість балів за ЗМ та модульний контроль	5			5			5			5			10	5		5		5		5		10	60	40				
Кількість балів за видами роботи	Лекції	Лабораторні	СРС	Лекції	Лабораторні	СРС	Лекції	Лабораторні	СРС	Лекції	Лабораторні	СРС	Модульний контроль	Лекції	Лабораторні	СРС	Лекції	Лабораторні	СРС	Лекції	Лабораторні	СРС	Модульний контроль					
	1			1			1			1				1			1			1				1				
		2			2			2			2					2			2			2				2		
			2			2			2			2					2				2					2		
Наукова робота	Участь у наукових конференціях, семінарах, круглих столах, студентських олімпіадах та конкурсах – 0-15 балів																						15					

Виконання курсового проекту (роботи)

Пояснювальна записка	Ілюстративна частина	Захист роботи	Сума
до 40	до 30	до 30	100

Критерії оцінювання модульного завдання

1. Правильна відповідь на питання модульного тесту з 1 по 20 оцінюється в 0,5 бала кожна.
2. Максимальна кількість балів за модульний контроль – 10.

Критерії оцінювання екзаменаційного завдання

Завдання	Кількість балів
Питання 1	10
Питання 2	10
Питання 3 (практичне завдання)	20
Разом	40

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

«Відмінно» - A (90-100 балів) – виставляється студенту, який глибоко та міцно засвоїв матеріал, відмінно справляється з задачами та питаннями, показує знайомство з фаховою літературою, володіє різносторонніми навичками та прийомами виконання практичних завдань, вміє добре орієнтуватись у виробничих ситуаціях.

«Добре» - BC (74-89 балів) – виставляється студенту, який твердо знає програмний матеріал, правильно застосовує теоретичні знання при рішенні практичних завдань, володіє необхідними навичками та прийомами їх виконання.

«Задовільно» - DE (64-73 балів) – виставляється студенту, який має знання тільки основного матеріалу, але не засвоїв його деталей, допускає неточності, неправильне тлумачення окремих елементів завдання та відчуває труднощі при виконанні практичних завдань.

«Незадовільно» - FX (35-59 балів) - виставляється студенту, який дає необґрунтовані відповіді на запитання, допускає суттєві помилки у використанні понятійного апарату. Не простежується логічність та послідовність думки. Формулювання хаотичні та не усвідомлені.

«Незадовільно» - F (1-34 балів) - виставляється студенту, який не засвоїв зміст дисципліни, вміння та навички не набуті.

13. Методичне забезпечення

1. Конспект лекцій з дисципліни «Крос-платформні технології в розробці інформаційних систем».
2. Методичні вказівки до виконання лабораторних робіт.
3. Методичні вказівки до виконання курсової роботи.
4. Методичні вказівки до виконання самостійних робіт.

14. Рекомендована література

1. Копитко М.Ф. Основи програмування мовою Java: Тексти лекцій / М. Ф. Копитко, К.С. Іванків. — Львів : Видавничий центр ЛНУ ім. Івана Франка, 2002. — 83 с.
2. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій / І. Є. Брнакевич, П. П. Вагін. — Львів : Видавничий центр ЛНУ імені Івана Франка, 2002. — 75 с.
3. Філіпс Б. Стюарт К. Марсикано К. Android. Програмування для професіоналів. 2-е видання. Київ: Діалектика, 2021. – 960 с.
4. Michael Wooldridge, An Introduction to MultiAgent Systems, John Wiley & Sons Ltd, 2002, paperback, 366 pages, ISBN 0-471-49691-X.
5. Carl Hewitt and Jeff Inman. DAI Betwixt and Between: From «Intelligent. Agents» to Open Systems Science IEEE Transactions on Systems, Man, and Cybernetics. Nov./Dec. 1991.
6. The Journal of Autonomous Agents and Multiagent Systems, Publisher: Springer Science+Business Media B.V., formerly Kluwer Academic Publishers B.V.

Електронні ресурси

1. <http://www.juce.com/>
2. <https://msdn.microsoft.com/uk-ua/>