

ЗАТВЕРДЖУЮ



Декан факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

09

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Крос-платформене програмування
Назва дисципліни

Призначення Робочої програми

Для освітніх програм різних спеціальностей

Рівень вищої освіти

Перший (бакалаврський)

Мова навчання

Українська

Обсяг дисципліни, кредитів ЄКТС

8

Статус дисципліни

Вибіркова

Факультет

Інформаційних технологій

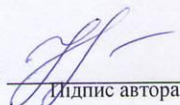
Кафедра

Комп'ютерних наук

Форма здобуття освіти	Загальний обсяг		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота, у т.ч. ІРС	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	68	32	36			172	+

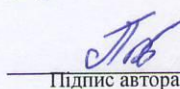
Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності

Робоча програма складена


Підпис автора

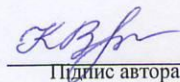
асистент каф. КН Ольга ЗАЛУЦЬКА

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора


Підпис автора

канд. пед. наук, доц. Сергій ПЕТРОВСЬКИЙ

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора


Підпис автора

асистент каф. КН Валерія КЛІМЕНКО

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора

Схвалена на засіданні кафедри Комп'ютерних наук

Протокол від 29.08.2025 № 1.

Зав. кафедри Комп'ютерних наук

Назва


Підпис

Олександр БАРМАК

Ім'я, ПРІЗВИЩЕ

ЛИСТ ПОГОДЖЕННЯ

Посада	Назва кафедри	Підпис	Ініціали, ПРИЗВИЩЕ
Завідувач кафедри, д-р. техн. наук, проф.	Комп'ютерних наук		Олександр БАРМАК
Гарант освітньо-професійної програми, канд. техн. наук, доц.	Комп'ютерних наук		Олександр МАЗУРЕЦЬ

3. Пояснювальна записка

Дисципліна «Крос-платформене програмування» спрямована на формування компетентностей у розробці програмного забезпечення для різних операційних систем і пристроїв. Вона забезпечує засвоєння архітектури крос-платформних додатків, стратегій інтеграції компонентів, веб-програмування (HTML, CSS, Javascript/Typescript), розробки клієнтських і мобільних застосунків, а також методів синхронної та асинхронної комунікації між компонентами системи. Вивчення дисципліни розвиває навички проектування та інтеграції програмних компонентів, забезпечення їхньої безпеки й ефективної роботи в різних середовищах..

Мета дисципліни. Формування у здобувачів знань і практичних навичок із розробки крос-платформного програмного забезпечення, здатності проектувати компоненти, інтегрувати їх у систему, створювати веб- та мобільні застосунки з урахуванням безпеки даних.

Предмет дисципліни. Принципи, методи та інструменти крос-платформного програмування, веб-орієнтованої розробки, інтеграції програмних компонентів та організації взаємодії між ними.

Завдання дисципліни. Набуття навичок проектування та реалізації крос-платформних додатків; розробки веб- та мобільних клієнтських застосунків із застосуванням HTML, CSS, Javascript/Typescript; використання Java та Java Spring Boot; забезпечення безпечного обміну даними між компонентами системи; опанування методів синхронної та асинхронної комунікації.

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен уміти проектувати програмні компоненти, інтегрувати їх у систему, розробляти веб- та мобільні застосунки на Java і Javascript/Typescript, використовувати Java Spring Boot для створення серверної частини, організовувати синхронну та асинхронну комунікацію між компонентами, забезпечувати захист інформації від несанкціонованого доступу та обслуговувати програмні системи в різних середовищах.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:		
	лекції	лабор. роботи	СРС
Тема 1. Крос-платформне програмування	2	4	16
Тема 2. Архітектура крос-платформного додатку, взаємодія крос-платформної та нативної частини.	2		
Тема 3. Стратегії інтеграції програмного забезпечення.	2	4	20
Тема 4. WEB-орієнтоване програмування.	4		
Тема 5. HTML розмітка, CSS та Javascript/Typescript як засіб відображення даних та роботи з ними.	4	4	36
Тема 6. Розробка клієнтських застосунків.	2	4	
Тема 7. Мобільні застосунки як продукт крос-платформного програмування.	2	4	20
Тема 8. Протоколи для комунікації між компонентами системи.	2		
Тема 9. Синхронні методи комунікації між компонентами.	2	4	36
Тема 10. Асинхронний обмін повідомленнями.	4	4	
Тема 11. Елементи програмування на Java; Java Spring Boot платформа.	2	4	44
Тема 12. Введення в Javascript/Typescript.	4	4	
Разом:	32	36	172

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Тема 1. Крос-платформне програмування</i>	2
1	Крос-платформне програмування. Поняття та основні визначення. Принципи крос-платформності. Переваги та недоліки підходу. Популярні середовища та інструменти. Приклади застосування. Літ.: [1, 2, 7, 8]	2
	<i>Тема 2. Архітектура крос-платформного додатку, взаємодія крос-платформної та нативної частини.</i>	2
2	Архітектура крос-платформного додатку, взаємодія крос-платформної та нативної частини. Компоненти архітектури. Взаємодія нативної та крос-платформної частини. Патерни побудови архітектури. Проблеми сумісності. Приклади архітектурних рішень. Літ.: [1, 2, 8]	2
	<i>Тема 3. Стратегії інтеграції програмного забезпечення.</i>	2
3	Стратегії інтеграції програмного забезпечення. Основні підходи до інтеграції. Вертикальна та горизонтальна інтеграція. API та сервіси. Контейнери та мікросервіси. Інтеграційне тестування. Літ.: [1, 4, 7]	2
	<i>Тема 4. WEB-орієнтоване програмування.</i>	4
4	WEB-орієнтоване програмування (частина 1). Клієнт-серверна архітектура. HTTP як основний протокол. Побудова простих веб-додатків. Інструменти для веб-розробника. Літ.: [12, 13, 14]	2
5	WEB-орієнтоване програмування (частина 2). Веб-фреймворки та бібліотеки. REST та SOAP сервіси. Архітектурні підходи до створення веб-додатків. Безпека в інтернет-середовищі. Тестування веб-систем. Літ.: [12, 13, 14]	2

	<i>Тема 5. HTML розмітка, CSS та Javascript/Typescript як засіб відображення даних та роботи з ними.</i>	4
6	HTML, CSS та основи Javascript. HTML як основа структури документа. Використання CSS для оформлення. Основні можливості Javascript. DOM і робота з подіями. Приклади інтеграції HTML, CSS і JS. Літ.: [4, 5, 6]	2
7	Javascript/Typescript як засіб роботи з даними. Типізація в Typescript. Моделі об'єктів та функцій. Робота з колекціями. Маніпуляції DOM. Подієва модель. Літ.: [4, 5, 6, 14]	2
	<i>Тема 6. Розробка клієнтських застосунків.</i>	2
8	Розробка клієнтських застосунків. Створення інтерфейсу користувача. Підключення та використання API. Використання бібліотек і фреймворків. Оптимізація продуктивності. Приклади реальних клієнтських додатків. Літ.: [1, 2, 7]	2
	<i>Тема 7. Мобільні застосунки як продукт крос-платформного програмування.</i>	2
9	Мобільні застосунки як продукт крос-платформного програмування. Підходи до створення мобільних додатків. React Native і Flutter. Інтеграція з нативними сервісами. Оптимізація під мобільні пристрої. Тестування мобільних рішень. Порівняння з нативними застосунками. Літ.: [7, 8]	2
	<i>Тема 8. Протоколи для комунікації між компонентами системи.</i>	2
10	Протоколи комунікації між компонентами. HTTP, WebSocket і gRPC. Синхронні та асинхронні моделі передачі даних. Проблеми сумісності протоколів. Приклади використання у різних середовищах. Літ.: [12, 14]	2
	<i>Тема 9. Синхронні методи комунікації між компонентами.</i>	2
11	Синхронні методи комунікації між компонентами. Виклики процедур і методів. Використання API у синхронному режимі. Приклади реалізації. Переваги та недоліки. Особливості використання в реальних проєктах. Літ.: [9, 11]	2
	<i>Тема 10. Асинхронний обмін повідомленнями.</i>	2
	Асинхронний обмін повідомленнями. Подієво-орієнтована модель. Асинхронні виклики у JS та Java. Обробка винятків. Приклади практичних сценаріїв. Проблеми асинхронної координації.	2
	<i>Тема 11. Елементи програмування на Java; Java Spring Boot платформа.</i>	4
	Елементи програмування на Java (частина 1). Основи синтаксису Java. Об'єктно-орієнтовані принципи. Структури даних. Колекції та обробка даних. Приклади програмних реалізацій. Літ.: [1, 8, 9]	2
	Елементи програмування на Java (частина 2). Основи Spring Boot. Конфігурація та структура проєкту. Робота з базами даних. Створення REST-сервісів. Тестування додатків. Приклади застосувань. Літ.: [1, 8, 9]	2
	<i>Тема 12. Введення в Javascript/Typescript.</i>	4
	Введення в Javascript. Синтаксис і базові оператори. Функції та об'єкти. Масиви та колекції. Події у Javascript. Приклади застосування у браузері. Літ.: [11, 12, 13]	2
	Введення в Typescript. Система типів. Класи та інтерфейси. Модулі й простори імен. Використання TS у веб-додатках. Приклади застосувань. Літ.: [12, 13]	2
	Разом:	32

5.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1	Знайомство з інструментарієм та введення в Java. Бібліотека Java. Літ.: [1, 2, 7]	4
2	Об'єктно орієнтоване програмування. Робота з файлами. Літ.: [1, 2]	4
3	Багатопоточне програмування. Робота з Stream API. Літ.: [9, 10]	4
4	Робота з елементами HTML. JavaScript Літ.: [4, 5, 6]	4
5	Побудова web застосунку. Створення веб сервісу. Літ.: [4, 5]	4
6	Підключення веб клієнта до Web API сервісу Літ.: [4, 5, 6]	4
7	Побудова мобільного застосунку за допомогою React Native Літ.: [13, 14]	4
8	Робота з нереляційними базами даних Літ.: [14]	4
Разом:		32

5.3 Зміст самостійної (у т.ч. індивідуальної) роботи здобувача вищої освіти

Самостійна робота студентів усіх форм здобуття освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до виконання і захисту лабораторних робіт та виконанні індивідуальних завдань. До послуг студентів сторінка кафедри у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні матеріали з її навчально-методичного забезпечення та контролю результатів навчання.

№ тижня	Вид самостійної роботи	Кількість годин
1-2	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1. Підготовка до захисту лабораторної роботи №1.	16
3-4	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №2. Підготовка до захисту лабораторної роботи №2.	16
5-6	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3. Підготовка до захисту лабораторної роботи №3. Підготовка до КР.	20
7-8	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №4. Підготовка до захисту лабораторної роботи №4. Початок виконання індивідуального завдання №1 самостійної роботи.	16
9-10	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5. Підготовка до захисту лабораторної роботи №5. Завершення виконання індивідуального завдання №1 самостійної роботи.	20
11-12	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6. Підготовка до захисту лабораторної роботи №6. Початок виконання індивідуального завдання №2 самостійної роботи.	20
13-14	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №7. Підготовка до захисту лабораторної роботи №7. Підготовка до КР.	20
15-16	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №8. Підготовка до захисту лабораторної роботи №8. Завершення виконання індивідуального завдання №2 самостійної роботи.	20
17	Опрацювання лекційного матеріалу. Підготовка звіту з самостійної роботи. Оформлення звіту з самостійної роботи. Презентація результатів виконання індивідуального завдання.	24
Разом		172

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до лабораторних занять та самостійної роботи індивідуальні завдання (ІЗ) з відповідних тем. Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання індивідуального завдання викладені в Модульному середовищі для навчання на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням інструктування, демонстрування, розв'язування типових і прикладних задач, аналізу кейсів, ситуаційних завдань, елементів дискусії тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, поточного та підсумкового контролю, виконання індивідуальних завдань), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

Поточний контроль здійснюється під час аудиторних лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, в т.ч. з використанням Модульного середовища для навчання. При цьому використовуються такі методи поточного контролю:

- оцінювання результатів виконання індивідуального завдання;
- оцінювання контрольних робіт;
- оцінювання результатів захисту лабораторних робіт.

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контролю, який проводиться з усього матеріалу дисципліни за білетами, попередньо розробленими і затвердженими на засіданні кафедри. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, **не допускається** до семестрового контролю, поки не виконає обсяг роботи, передбачений Робочою програмою. Здобувач вищої освіти, який набрав позитивний середньозважений бал (60 відсотків і більше від максимального балу) з усіх видів поточного контролю і не склав іспит, вважається таким, який **має** академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в он-лайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до лабораторного заняття (вивчення теоретичного матеріалу з теми роботи, попередню підготовку протоколу роботи (наведені у Методичних рекомендаціях до лабораторних занять)), активно працювати на занятті, якісно підготувати звіт, захистити результати виконаної роботи, брати участь у дискусіях щодо прийнятих конструктивних рішень при виконанні здобувачами лабораторних робіт тощо.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати у

встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час лабораторних занять, виконання індивідуального завдання та контрольних робіт. Виконання індивідуального завдання завершується його здачею на перевірку у терміни, встановлені графіком самостійної роботи.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (в т.ч. із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності **не допускаються**.

У межах вивчення навчальної дисципліни допускається визнання та зарахування результатів навчання, здобутих шляхом неформальної освіти на відкритих освітніх платформах. Такі результати можуть враховуватися за умови їх відповідності компетентностям і програмним результатам навчання, визначеним робочою програмою дисципліни, або у разі забезпечення вивчення окремих тем і видів робіт, що передбачені програмою. (детальніше у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. При цьому кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти 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 мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>похибки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, 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 правил, закономірностей тощо. Відповідь

	здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три несуттєві помилки .
Задовільно (достатній)	Здобувач вищої освіти 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 здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, 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 (недостатній)	Здобувач вищої освіти 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 здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота								Контрольні заходи		Самостійна робота		Семестровий контроль
Лабораторні заняття №:								Контрольна робота (КР)		Індивідуальне завдання (ІЗ)		Залік
1	2	3	4	5	6	7	8	КР1	КР2	ІЗ1	ІЗ2	
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)												
6-10	6-10	6-10	6-10	6-10	6-10	6-10	6-10	3-5	3-5	3-5	3-5	За рейтингом
48-80								6-10		6-10		60-100*

Примітки: *За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти.

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому **не зараховується** і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання контрольної роботи

Контрольна робота передбачає виконання п'яти завдань. При оцінюванні контрольної роботи враховуються: повнота відповіді та якість виконання. Кожне завдання оцінюється 1 балом, загальна сума балів на позитивну оцінку становить від 3 до 5.

Розподіл балів при оцінюванні завдань контрольної роботи

Кількість правильних відповідей	1	2	3	4	5
Відсоток правильних відповідей	0-59		60	80	100
Кількість отриманих балів	0		3	4	5

При отриманні негативної оцінки контрольну роботу слід перездати до терміну **наступного** контролю.

Оцінювання результатів виконання індивідуального завдання

Виконане та оформлене відповідно до вимог, визначених Методичними рекомендаціями, індивідуальне завдання (ІЗ) комплексно оцінюється викладачем з урахуванням таких критеріїв: самостійність виконання; правильність розв'язання поставлених задач; обґрунтованість вибору методів розв'язання; повнота пояснень та аргументованість відповідей; якість оформлення та дотримання вимог до структури і змісту роботи.

Результат виконання здобувачем вищої освіти кожного ІЗ оцінюється відповідно до таблиці **Критеріїв оцінювання навчальних досягнень здобувача вищої освіти з урахуванням рівня досягнення запланованих програмних результатів навчання та сформованих компетентностей**. За підсумками захисту присвоюється відповідна сума балів (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У разі, якщо здобувач вищої освіти виявив рівень знань і виконання ІЗ, що нижчий ніж 60 відсотків від максимальної кількості балів, встановленої Робочою програмою для цієї структурної одиниці, завдання не зараховується. У такому випадку студент має повторно опрацювати зміст завдання, усунути помилки та здати на перевірку доопрацьоване ІЗ у терміни, погоджені з викладачем.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий залік виставляється на останньому занятті за умови якщо загальна сума балів, яку накопичив здобувач з дисципліни (іншого освітнього компонента) за результатами **поточного** контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «**зараховано**», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення. Присутність здобувача у цьому випадку не є обов'язковою.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		Незадовільно/Fail – Результати навчання відсутні

10. Питання для самоконтролю результатів навчання

1. Що таке крос-платформне програмування і які його ключові принципи?

2. Які переваги та недоліки має крос-платформність у розробці програмного забезпечення?
3. Які інструменти та середовища найчастіше використовуються для крос-платформної розробки?
4. Як крос-платформність впливає на продуктивність програмного забезпечення?
5. Які приклади успішних крос-платформних проєктів ви знаєте?
6. З яких компонентів зазвичай складається архітектура крос-платформного додатку?
7. Як відбувається взаємодія між нативною та крос-платформною частиною застосунку?
8. Які патерни проєктування застосовуються у побудові архітектури крос-платформних додатків?
9. У чому полягають основні проблеми сумісності між платформами?
10. Наведіть приклади архітектурних рішень для різних типів застосунків.
11. Які існують основні стратегії інтеграції програмного забезпечення?
12. У чому полягає різниця між вертикальною та горизонтальною інтеграцією?
13. Як API сприяють інтеграції різних компонентів системи?
14. Яку роль відіграють контейнери та мікросервіси у сучасній інтеграції?
15. Які методи інтеграційного тестування ви знаєте?
16. Що таке клієнт–серверна архітектура і як вона працює?
17. Які основні властивості протоколу HTTP?
18. Як побудувати простий веб-додаток?
19. Які інструменти використовуються веб-розробниками на початкових етапах?
20. Які основні виклики виникають у процесі веб-орієнтованого програмування?
21. Які відмінності між REST та SOAP сервісами?
22. Які веб-фреймворки є найбільш поширеними?
23. У чому полягають основні архітектурні підходи до побудови веб-додатків?
24. Які аспекти безпеки є найважливішими при веб-розробці?
25. Як здійснюється тестування веб-систем?
26. Яке призначення HTML у побудові веб-сторінки?
27. Які можливості надає CSS для оформлення контенту?
28. Як за допомогою Javascript можна взаємодіяти з елементами DOM?
29. У чому полягає подієва модель у Javascript?
30. Як інтегруються HTML, CSS і JS у веб-додатках?
31. Які особливості типізації в Typescript?
32. Як у Javascript/Typescript організовано роботу з функціями?
33. Які підходи існують до роботи з колекціями?
34. Як здійснюються маніпуляції з DOM у сучасних веб-додатках?
35. Яку роль відіграє подієва модель у розробці?
36. Які етапи передбачає створення інтерфейсу користувача?
37. Як клієнтські застосунки використовують API?
38. Які фреймворки найчастіше застосовуються для клієнтської розробки?
39. Як забезпечити оптимізацію продуктивності клієнтського застосунку?
40. Наведіть приклади клієнтських додатків у реальних проєктах.
41. Які підходи існують до створення мобільних застосунків?
42. У чому полягають особливості React Native та Flutter?
43. Як мобільні застосунки інтегруються з нативними сервісами пристроїв?
44. Які методи оптимізації використовуються для мобільних застосунків?
45. У чому відмінність між крос-платформними та нативними мобільними застосунками?
46. Які протоколи комунікації найчастіше застосовуються у розподілених системах?
47. У чому особливість WebSocket у порівнянні з HTTP?
48. Як працює протокол gRPC?
49. Які проблеми сумісності виникають між різними протоколами?
50. Як вибрати оптимальний протокол для конкретного проєкту?
51. Що таке синхронний виклик процедур?
52. Як працює виклик API у синхронному режимі?
53. Які приклади реалізації синхронних методів відомі?
54. Які переваги та недоліки має синхронна комунікація?
55. Як синхронні методи застосовуються у реальних проєктах?

56. Що таке подієво-орієнтована модель у програмуванні?
57. Як реалізуються асинхронні виклики у Javascript?
58. Які підходи існують до обробки винятків в асинхронних сценаріях?
59. Які проблеми виникають під час асинхронної координації компонентів?
60. Наведіть приклади застосування асинхронних методів у сучасних системах.
61. Які базові елементи синтаксису Java ви знаєте?
62. Як реалізуються принципи ООП у Java?
63. Які структури даних доступні у Java?
64. Як працюють колекції у Java?
65. Наведіть приклади програмних реалізацій основних структур даних.
66. Які основні можливості надає Spring Boot?
67. Як здійснюється конфігурація та організація проєкту на Spring Boot?
68. Які методи роботи з базами даних доступні у Spring Boot?
69. Як створюються REST-сервіси за допомогою Spring Boot?
70. Які інструменти застосовуються для тестування Java-додатків?
71. Які основні оператори використовуються у Javascript?
72. Як визначаються функції та об'єкти у Javascript?
73. Які можливості надають масиви у Javascript?
74. Як працює система подій у Javascript?
75. Наведіть приклади застосування JS у браузері.
76. Які ключові особливості системи типів у Typescript?
77. Як у Typescript реалізуються класи та інтерфейси?
78. Що таке модулі у Typescript і для чого вони потрібні?
79. Як TS інтегрується у веб-додатки?
80. Наведіть приклади застосування Typescript у реальних проєктах.

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Крос-платформене програмування» забезпечений необхідними навчально-методичними матеріалами, що розміщені в Модульному середовищі для навчання MOODLE:

1. Курс «Крос-платформене програмування».

<https://msn.khmnpu.edu.ua/course/view.php?id=432>

12. Матеріально-технічне та програмне забезпечення дисципліни

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проєктор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями.

Для виконання практичних завдань необхідним є застосування безкоштовних платформ та інструментів, зокрема Visual Studio Code як середовища розробки, а також GitHub для контролю версій та командної роботи. Додатково використовуються фреймворки та середовища для крос-платформної розробки, такі як React Native та Flutter, а також середовище Node.js для роботи з JavaScript/TypeScript і платформа Spring Boot для програмування на Java.

13. Рекомендована література:

Основна

1. Кросплатформна розробка мобільних застосунків [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спец. 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського ; уклад.: О. Л. Недашківський, І. І. Гусева. – Київ : КПІ ім. Ігоря Сікорського, 2023. – 221 с. – Електронне видання (PDF). – Режим доступу: <https://ela.kpi.ua/items/f9d71363-d4c6-4979-8621-75a3a2608cdb>

2. Haverbeke M. Eloquent JavaScript. 4th ed. – No Starch Press, 2024. – 490 p. – Електронне видання (PDF). – Режим доступу: <https://eloquentjavascript.net/>

3. The TypeScript Handbook. – Microsoft, 2023. – 330 p. – Електронне видання (PDF). – Режим доступу: <https://www.typescriptlang.org/docs/handbook/intro.html>
4. Basarat A. TypeScript Deep Dive. – Leanpub, 2020. – 550 p. – Електронне видання (PDF). – Режим доступу: <https://basarat.gitbook.io/typescript/>
5. JavaScript Notes for Professionals. – GoalKicker, 2022. – 400 p. – Електронне видання (PDF). – Режим доступу: <https://goalkicker.com/JavaScriptBook/>
6. Eland M. Uno Platform Succinctly. – Syncfusion, 2021. – 145 p. – Електронне видання. URL: <https://www.syncfusion.com/succinctly-free-ebooks/uno-platform-succinctly>

Допоміжна

7. Кросплатформне програмування. Матеріали до лекцій. – Електронне видання (PDF). – Режим доступу: <https://labs.starbasic.net/krosplatformne-prohramuvannia/>
8. Крос-платформне програмування: методичні вказівки та завдання до виконання лабораторних робіт / уклад.: В.М. Хроленко, В.Г. Голенков. – К.: КНУБА, 2024. – 15 с. – Електронне видання (PDF). – Режим доступу: <https://repository.knuba.edu.ua/bitstreams/be3022e2-de64-4a4d-bbc9-04b16a35aa76/download>.
9. Oracle. *The Java® Language Specification*. – 8th ed. – Oracle Corporation, 2015. – 765 с. – Електронне видання (PDF). – Режим доступу: <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>
10. Svekis, L., van Putten, M., Percival, R. *JavaScript from Beginner to Professional*. – Birmingham: Packt Publishing, 2021. – 350 с. – Електронне видання (PDF). – Режим доступу: <https://www.gurukultti.org/admin/notice/javascript.pdf>
11. Flavio Copes. *The JavaScript Handbook*. – 2025. – 200 с. – Електронне видання (PDF). – Режим доступу: <https://www.edu-9.de/uploads/books/javascript-handbook.pdf>
12. Mozilla Developer Network. *JavaScript Guide*. – Mozilla Foundation, 2025. – 200 с. – Електронне видання (HTML). – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
13. Paul Fodor. *HTML and CSS*. – Stony Brook University, 2018. – 100 с. – Електронне видання (PDF). – Режим доступу: https://www3.cs.stonybrook.edu/~pfodor/courses/CSE316/L03-HTML_CSS.pdf
14. Eloquent JavaScript. – 4th ed. – 2025. – 500 с. – Електронне видання (PDF). – Режим доступу: https://eloquentjavascript.net/Eloquent_JavaScript_small.pdf

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=432>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	—
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен уміти проектувати програмні компоненти, інтегрувати їх у систему, розробляти веб- та мобільні застосунки на Java і Javascript/Typescript, використовувати Java Spring Boot для створення серверної частини, організовувати синхронну та асинхронну комунікацію між компонентами, забезпечувати захист інформації від несанкціонованого доступу та обслуговувати програмні системи в різних середовищах.

Зміст навчальної дисципліни. Крос-платформне програмування. Архітектура крос-платформного додатку, взаємодія крос-платформної та нативної частини. Стратегії інтеграції програмного забезпечення. WEB-орієнтоване програмування. HTML розмітка, CSS та Javascript/Typescript як засіб відображення даних та роботи з ними. Розробка клієнтських застосунків. Мобільні застосунки як продукт крос-платформного програмування. Протоколи для комунікації між компонентами системи. Синхронні методи комунікації між компонентами. Асинхронний обмін повідомленнями. Елементи програмування на Java; Java Spring Boot платформа. Введення в Javascript/Typescript.

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для *першого* (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням інструктування, демонстрування, розв'язування типових і прикладних задач, аналізу кейсів, ситуативних завдань, елементів дискусії тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, поточного та підсумкового контролю, виконання індивідуальних завдань), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: оцінювання лабораторних та контрольних робіт; виконання індивідуального завдання.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Кросплатформна розробка мобільних застосунків [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спец. 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського ; уклад.: О. Л. Недашківський, І. І. Гусева – Київ : КПІ ім. Ігоря Сікорського, 2023. – 221 с. – Електронне видання (PDF). – Режим доступу: <https://ela.kpi.ua/items/f9d71363-d4c6-4979-8621-75a3a2608cdb>
2. Haverbeke M. Eloquent JavaScript. 4th ed. – No Starch Press, 2024. – 490 p. – Електронне видання (PDF). – Режим доступу: <https://eloquentjavascript.net/>
3. The TypeScript Handbook. – Microsoft, 2023. – 330 p. – Електронне видання (PDF). – Режим доступу: <https://www.typescriptlang.org/docs/handbook/intro.html>
4. Basarat A. TypeScript Deep Dive. – Leanpub, 2020. – 550 p. – Електронне видання (PDF). – Режим доступу: <https://basarat.gitbook.io/typescript/>
5. JavaScript Notes for Professionals. – GoalKicker, 2022. – 400 p. – Електронне видання (PDF). – Режим доступу: <https://goalkicker.com/JavaScriptBook/>
6. Eland M. Uno Platform Succinctly. – Syncfusion, 2021. – 145 p. – Електронне видання. URL: <https://www.syncfusion.com/succinctly-free-ebooks/uno-platform-succinctly>
7. Кросплатформне програмування. Матеріали до лекцій. – Електронне видання (PDF). – Режим доступу: <https://labs.starbasic.net/krosplatformne-prohramuvannia/>

8. Крос-платформне програмування: методичні вказівки та завдання до виконання лабораторних робіт / уклад.: В.М. Хроленко, В.Г. Голенков. – К.: КНУБА, 2024. – 15 с. – Електронне видання (PDF). – Режим доступу: <https://repository.knuba.edu.ua/bitstreams/be3022e2-de64-4a4d-bbc9-04b16a35aa76/download>.
9. Oracle. *The Java® Language Specification*. – 8th ed. – Oracle Corporation, 2015. – 765 с. – Електронне видання (PDF). – Режим доступу: <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>
10. Svekis, L., van Putten, M., Percival, R. *JavaScript from Beginner to Professional*. – Birmingham: Packt Publishing, 2021. – 350 с. – Електронне видання (PDF). – Режим доступу: <https://www.gurukuliti.org/admin/notice/javascript.pdf>
11. Flavio Copes. *The JavaScript Handbook*. – 2025. – 200 с. – Електронне видання (PDF). – Режим доступу: <https://www.edu-9.de/uploads/books/javascript-handbook.pdf>
12. Mozilla Developer Network. *JavaScript Guide*. – Mozilla Foundation, 2025. – 200 с. – Електронне видання (HTML). – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
13. Paul Fodor. *HTML and CSS*. – Stony Brook University, 2018. – 100 с. – Електронне видання (PDF). – Режим доступу: https://www3.cs.stonybrook.edu/~pfodor/courses/CSE316/L03-HTML_CSS.pdf
14. Eloquent JavaScript. – 4th ed. – 2025. – 500 с. – Електронне видання (PDF). – Режим доступу: https://eloquentjavascript.net/Eloquent_JavaScript_small.pdf
15. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=432>
16. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
17. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

Викладачі: асистент, Залуцька О.О.
к.п.н., доцент, Петровський О.В.
асистент, Кліменко С.С..