

ЗАТВЕРДЖУЮ

Декан факультету інформаційних технологій



Тетяна ГОВОРУЩЕНКО

Підпис Ім'я, ПРІЗВИЩЕ

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мобільно-орієнтована розробка програмного забезпечення

Назва дисципліни

Призначення Робочої програми

Для освітніх програм різних спеціальностей

Рівень вищої освіти

Перший (бакалаврський)

Мова навчання

Українська

Обсяг дисципліни, кредитів ЕКТС

8

Статус дисципліни

Вибіркова

Факультет

Інформаційних технологій

Кафедра

Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	98	16	64	18		142	+

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

Робоча програма складена

ст.викладач Дмитро ДЕНИСЮК

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ

Схвалена на засіданні кафедри

комп'ютерної інженерії та інформаційних систем

Назва

Протокол від 18.08.2025 № 1.

Зав. кафедри комп'ютерної інженерії та інформаційних систем

Назва

Ольга ПАВЛОВА

Ім'я, ПРІЗВИЩЕ

3. Пояснювальна записка

Вибіркова навчальна дисципліна «Мобільно-орієнтована розробка програмного забезпечення» реалізує принципи індивідуалізації освітньої траєкторії та забезпечує формування спеціалізованих професійних компетентностей у галузі сучасного мобільного програмування. Вивчення дисципліни сприяє системній інтеграції теоретичних знань з практичними навичками розробки нативних мобільних додатків з використанням фреймворку React Native, що відповідає актуальним вимогам IT-ринку та сучасним тенденціям розвитку програмної інженерії.

Мета дисципліни. Формування у здобувачів вищої освіти цілісної системи теоретичних знань, практичних умінь та професійних компетентностей у галузі мобільно-орієнтованої розробки програмного забезпечення, що передбачає оволодіння сучасними концепціями, методами та інструментальними засобами створення, тестування й оптимізації нативних і кросплатформених мобільних застосунків із використанням мови програмування JavaScript та фреймворку React Native. Дисципліна спрямована на розвиток здатності здобувачів до науково обґрунтованого вибору технологічних рішень, застосування сучасних фреймворків і середовищ розробки, інтеграції програмних компонентів та ефективного вирішення прикладних завдань у сфері мобільної інженерії.

Предмет дисципліни. Методологічні засади та практичні аспекти розроблення нативних мобільних застосунків на основі інтеграції засобів мови програмування JavaScript із технологічним стеком фреймворку React Native, реалізованих на прикладі платформи Android. Предметом вивчення є архітектурні принципи побудови мобільних програмних систем, компонентно-орієнтований підхід до проєктування інтерфейсів і логіки застосунків, методи управління їхнім станом, а також особливості взаємодії з апаратними ресурсами та сервісами мобільних пристроїв.

Завдання дисципліни. Формування у здобувачів вищої освіти практичних умінь і навичок розроблення мобільних застосунків із використанням фреймворку React Native та мови програмування JavaScript. Опанування принципів проєктування та реалізації користувацьких інтерфейсів, організації взаємодії з даними, подіями та зовнішніми сервісами. Засвоєння методів тестування, налагодження, оптимізації та публікації мобільних застосунків у середовищах розгортання. Розвиток аналітичного та системного мислення, здатності до командної взаємодії, прийняття обґрунтованих інженерних рішень і застосування сучасних технологічних підходів у сфері мобільної інженерії.

Результати навчання. Після успішного завершення курсу здобувач вищої освіти демонструє комплексне засвоєння теоретичних основ та практичних аспектів розробки мобільних застосунків. Студент володіє системними знаннями архітектурних принципів фреймворку React Native, включаючи компонентну модель, роботу з JSX, управління станами (State) та властивостями (Props), а також особливості створення нативних мобільних рішень. У практичній площині сформовані вміння ефективного використання інструментів мови JavaScript та сучасного середовища розробки для реалізації крос-платформових проєктів. Здобувач освіти вміє проєктувати архітектуру додатків, реалізовувати функціональність відповідно до технічного завдання, проводити комплексне тестування та обґрунтовувати вибір технологічних рішень. Особливу увагу приділено розвитку дослідницьких компетенцій, зокрема роботі з технічною документацією, аналізу сучасних тенденцій розробки та оптимізації продуктивності застосунків в умовах обмежених системних ресурсів. Соціально-професійний розвиток включає формування здатності до ефективної командної взаємодії, адаптивного самостійного навчання, ясної професійної комунікації та прийняття відповідальних інженерних рішень у відповідності до вимог сучасного IT-ринку.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
Тема 1. Вступ до мобільно-орієнтованої розробки та основи TypeScript	2	8	2	17
Тема 2. Основи роботи з React Native та середовище Expo.	2	8	2	17
Тема 3. Компоненти, стилізація та побудова інтерфейсів користувача	2	8	2	17
Тема 4. Навігація та взаємодія з користувачем	2	8	2	17
Тема 5. Управління станом застосунку із застосуванням Zustand	2	8	2	17
Тема 6. Робота з API, асинхронним кодом і зовнішніми сервісами	2	8	4	19
Тема 7. Зберігання та управління даними: Supabase та Appwrite	2	8	2	19
Тема 8. Тестування, оптимізація та публікація мобільних застосунків	2	8	2	19
Разом за семестр:	16	64	18	142

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1	Лекція 1. Вступ до мобільно-орієнтованої розробки та основи TypeScript	2
1.1	Концепції мобільної розробки. Види мобільних застосунків: нативні, гібридні, кросплатформенні. Переваги та обмеження підходів. Архітектура мобільних застосунків і вимоги до сучасного середовища розробки.	1
1.2	Вступ до TypeScript як основної мови для React Native. Типізація, інтерфейси, дженерики, модулі. Інтеграція TypeScript у React Native-проект. Переваги типобезпечної розробки у великих застосунках.	1
2	Лекція 2. Основи роботи з React Native та середовище Expo	2
2.1	Архітектура фреймворку React Native. Компонентна модель, JSX, props, state, render. Огляд інструментів розробки: Node.js, npm, Android Studio, VS Code. Створення першого застосунку.	1
2.2	Expo — швидкий старт у розробці мобільних застосунків. Порівняння Expo та нативного підходу. Переваги, обмеження та сценарії використання. Створення, запуск і деплой Expo-проекту.	1
3	Лекція 3. Компоненти, стилізація та побудова інтерфейсів користувача	2
3.1	Базові та користувацькі компоненти React Native. Життєвий цикл компонентів, побудова UI з використанням компонентного підходу. Організація структури застосунку.	1
3.2	Стилізація інтерфейсів у React Native. Використання StyleSheet, styled-components, Tailwind RN. Принципи адаптивного дизайну. Побудова макетів, гнучка верстка, позиціонування.	1
4	Лекція 4. Навігація та взаємодія з користувачем	2
4.1	Навігаційні бібліотеки React Native. Робота з React Navigation: Stack, Tab, Drawer навігація. Управління потоками екранів, передача параметрів, стан навігації.	1
4.2	Взаємодія користувача з інтерфейсом. Оброблення подій, дотиків і жестів. Реалізація сенсорних елементів, анімації та переходів. Використання react-native-gesture-handler і Animated API.	1
5	Лекція 5. Управління станом застосунку із застосуванням Zustand	2
5.1	Концепція стану у React. Проблеми глобального стану. Огляд бібліотек керування станом: Redux, Context, Zustand. Переваги Zustand State Manager.	1
5.2	Реалізація глобального стану з використанням Zustand. Створення store, selectors, middleware, підключення до компонентів. Приклади організації стану для реальних застосунків.	1
6	Лекція 6. Робота з API, асинхронним кодом і зовнішніми сервісами	2

6.1	Організація асинхронного коду в React Native. Використання Promise, async/await, обробка помилок. Взаємодія з RESTful API, методи запитів, HTTP-бібліотеки (Axios, Fetch).	1
6.2	Інтеграція зовнішніх сервісів. Аутентифікація користувачів через OAuth, робота з API сторонніх платформ. Кешування, локальне сховище, офлайн-доступ до даних.	1
7	Лекція 7. Зберігання та управління даними: Supabase та Appwrite	2
7.1	Використання Supabase у мобільних застосунках. Архітектура, аутентифікація, робота з базами даних PostgreSQL. CRUD-операції, реальний час, підключення до React Native.	1
7.2	Платформа Appwrite як альтернатива Firebase. Серверна архітектура, API, автентифікація користувачів, функції, сховище. Порівняння Supabase і Appwrite у контексті мобільної розробки.	1
8	Лекція 8. Тестування, оптимізація та публікація мобільних застосунків	2
8.1	Методи тестування мобільних застосунків. Unit-тести, інтеграційні тести, E2E-тестування. Використання Jest, React Native Testing Library. Налаштування, аналіз продуктивності, профілювання.	1
8.2	Підготовка до релізу. Оптимізація коду, збирання застосунку (Expo build, EAS Build, Android Studio). Публікація у Google Play та App Store. Підтримка, оновлення та масштабування.	1
Разом за семестр:		16

5.2 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
1	Налаштування середовища та початок роботи з TypeScript і Expo. Вибір середовища розробки. Створення Expo-проєкту та основи TypeScript.	2
2	Архітектура React Native та Базові компоненти. Створення UI з використанням JSX, props, state.	2
3	Стилізація компонентів та адаптивний дизайн. Використання StyleSheet та основ styled-components/Tailwind RN.	2
4	Робота з API. Асинхронний код та HTTP-запити. Використання Promise, async/await для взаємодії з RESTful API (на прикладі Fetch/Axios).	2
5	Реалізація користувацького компонента та обробка жестів. Створення компонента з використанням react-native-gesture-handler і анімації.	2
6	Керування навігаційним потоком (React Navigation). Реалізація Stack, Tab або Drawer навігації та передача параметрів.	2
7	Управління станом із Zustand. Створення глобального store (сховища) та підключення його до компонентів.	2
8	Робота з Supabase/Appwrite. Аутентифікація користувача, читання та запис даних.	4
Разом за семестр		18

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
1	Вступ до TypeScript та Expo. Налаштування середовища розробки. Типізація даних, інтерфейси та дженерики в мобільному проєкті. Створення проєкту на основі Expo	8
2	Компонентна модель React Native. Базові та користувацькі компоненти. Робота з props, state та життєвим циклом компонентів.	8
3	Сучасна стилізація інтерфейсів. Застосування StyleSheet та інтеграція бібліотек (styled-components або Tailwind RN). Побудова макетів та адаптивний дизайн.	8
4	Навігація за допомогою React Navigation. Реалізація різних видів навігації: Stack, Tab, Drawer. Передача параметрів між екранами.	8
5	Обробка взаємодії та API. Оброблення дотиків, жестів (react-native-gesture-handler) та основи анімацій.	8
6	Управління станом із Zustand. Створення Store, підключення до компонентів, використання selectors та middleware.	8
7	Інтеграція Supabase/Appwrite (Аутентифікація). Налаштування Backend-сервісу. Реалізація реєстрації та автентифікації користувачів.	8
8	Робота з даними Supabase/Appwrite. Виконання CRUD-операцій (створення, читання, оновлення, видалення) даних у NoSQL/PostgreSQL базі.	8
Разом:		64

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, підготовці до виконання і захисту лабораторних робіт, підготовці до практичних занять, тестуванні з теоретичного матеріалу, виконанні індивідуальних завдань тощо. До послуг студентів сторінка кафедри у Модульному середовищі для навчання MOODLE, де розміщені Робоча програма дисципліни та необхідні матеріали з її навчально-методичного забезпечення та контролю результатів навчання (зокрема, тестові завдання для поточного та семестрового контролю знань).

Номер тижня	Вид самостійної роботи	Кількість годин
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1 та ПЗ1. Підготовка до захисту ЛР1.	17
3-4	Опрацювання лекційного матеріалу. Підготовка до ЛР2 та ПЗ2. Підготовка до захисту ЛР2.	17
5-6	Опрацювання лекційного матеріалу. Підготовка до ЛР3 та ПЗ3. Підготовка до захисту ЛР3.	17
7-8	Опрацювання лекційного матеріалу. Підготовка до ЛР4 та ПЗ4. Підготовка до захисту ЛР4.	17
9-10	Опрацювання лекційного матеріалу. Підготовка до ЛР5 та ПЗ5. Підготовка до захисту ЛР5.	17
11-12	Опрацювання лекційного матеріалу. Підготовка до ЛР6 та ПЗ6. Підготовка до захисту ЛР6.	19
13-14	Опрацювання лекційного матеріалу. Підготовка до ЛР7 та ПЗ7. Підготовка до захисту ЛР7.	19
15-16	Опрацювання лекційного матеріалу. Підготовка до ЛР8 та ПЗ8. Підготовка до захисту ЛР8.	19
Разом 1-й семестр:		142

Примітки: ЛР – лабораторна робота, ПЗ– практичне заняття.

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до практичних або лабораторних занять та самостійної роботи питання з кожної теми. Керівництво самостійною роботою та контроль за опрацюванням матеріалу здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема:

- **Лекції** з використанням мультимедійних презентацій, методів візуалізації, проблемного й інтерактивного навчання, методів стимулювання і мотивації, інформаційно-комунікаційних технологій.
- **Практичні заняття** з використанням інструктування, демонстрування, розв'язування ситуаційних завдань (кейсів), елементів дискусії та мозкового штурму.
- **Лабораторні заняття** з використанням методів комп'ютерного моделювання, проектної діяльності, тренінгових вправ та аналізу проблемних ситуацій, що забезпечує формування практичних навичок.

7. Методи контролю

Поточний контроль знань студентів здійснюється протягом семестру під час аудиторних (практичних та лабораторних) занять, а також у дні проведення контрольних заходів, встановлених робочою програмою, в тому числі з використанням Модульного середовища для навчання MOODLE. При цьому використовуються такі методи поточного контролю:

- Оцінювання результатів роботи на практичних заняттях оцінка усного опитування теоретичного матеріалу, розв'язування прикладних задач та участь у обговоренні кейсів і ситуаційних завдань.
- Усне опитування перед допуском до виконання лабораторної роботи перевірка засвоєння необхідного теоретичного матеріалу.
- Оцінювання результатів захисту лабораторних робіт перевірка якості виконання роботи та набуття практичних навичок.
- Тестовий контроль перевірка засвоєння теоретичного та практичного матеріалу з тем (розділів) дисципліни.

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, 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. Проходження інструктажу з техніки безпеки та 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 та формування фахових компетентностей передбачає необхідність підготовки до практичних та лабораторних занять, що включає вивчення теоретичного матеріалу з теми та попередню підготовку до усного опитування для допуску до заняття. Важливо активно працювати на занятті та брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт. До видів робіт, за які студенти будуть отримувати оцінки, належать оцінювання результатів роботи на практичних заняттях, оцінювання результатів захисту лабораторних робіт та тестовий контроль теоретичного і практичного матеріалу.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене практичне і 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 усного опитування під час практичних занять та тестування.

Здобувач вищої освіти, виконуючи самостійну або індивідуальну роботу, має дотримуватися політики доброчесності; заборонені списування та плагіат, в тому числі з 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 здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених робочою програмою дисципліни (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 <i>несуттєві помилки</i> .
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 <i>суттєві помилки</i> 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	Семестровий контроль		
Лабораторні заняття №:								Практичні заняття				Тестовий контроль	Залік		
1*	2	3	4	5	6	7	8	1	2	3	4	Т 1-2			
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)															
5-8	5-8	5-8	5-8	5-8	5-8	5-8	5-8	3-4	3-4	3-4	3-4	5–10			
40–64								12–16				10–20		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 роботи на практичних заняттях

Оцінка, яка виставляється за практичне заняття (№ 1-4), складається з таких елементів: усне опитування студентів на знання теоретичного матеріалу з теми; вільне володіння професійною термінологією та вміння професійно обґрунтувати прийняті рішення при розв'язуванні прикладних задач; результати самостійних робіт. При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «Критерії оцінювання навчальних досягнень здобувача вищої освіти» (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота (№ 1-8) комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання методики розроблення мобільного застосунку (проектування); вміння професійно обґрунтувати прийняті технологічні рішення.

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. 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 результатів тестового контролю

Протягом семестру проводяться два тестування, які перевіряють рівень засвоєння теоретичного матеріалу та здатність застосовувати знання на практиці. Кожен тест складається із 20 завдань однакової складності. Залежно від кількості правильних відповідей студент може отримати від 5 до 10 балів за тест. Таким чином, загальний результат тестового контролю за семестр становить від 10 до 20 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	0-11	12-13	14	15-16	17	18-20
Відсоток правильних відповідей	0-59	60-65	66-72	73-82	83-89	90-100
Кількість отриманих балів	0	6	7	8	9	10

На тестування відводиться 25 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. Також, студент може проходити тестування письмово, записуючи правильні відповіді у талоні відповідей. При отриманні негативної оцінки тест слід перездати до терміну *наступного* контролю.

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		
D	66-72		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
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. Який вид мобільних застосунків створюється за допомогою React Native?
2. Яка основна архітектура використовується у React Native?
3. Яку мову програмування обрано основною для розробки у новій програмі?
4. Яке ключове поняття в TypeScript дозволяє визначати форму об'єктів?
5. Як називається механізм у TypeScript, що дозволяє створювати повторно використовувані компоненти, незалежні від конкретних типів?
6. Який інструмент використовується для швидкого старту та збирання мобільних застосунків у новій програмі?
7. Що забезпечує Expo для прискорення процесу тестування застосунку на реальному пристрої?
8. Як називається синтаксичне розширення JavaScript, що використовується для опису UI у React Native?
9. Яка технологія забезпечує кросплатформність у React Native, дозволяючи JS-коду взаємодіяти з нативними модулями?
10. Назвіть середовище розробки, рекомендоване для React Native-проектів.
11. Який інструмент керує пакетами та залежностями у React Native-проекті?
12. Що таке типобезпечна розробка?
13. Який елемент архітектури React Native відповідає за перетворення JS-викликів у нативні виклики?
14. Як називається набір інструментів, що дозволяє створювати, запускати та публікувати Expo-проекти?
15. Який інструмент використовується для створення нативного білда проекту без Expo?
16. Як називається підхід до розробки, коли один і той самий код використовується для iOS та Android?
17. Що таке Нативні застосунки?
18. Як називається процес, коли застосунок вивантажується на пристрій для тестування?
19. Що таке інтерфейси в TypeScript?
20. Який компонент у React Native відповідає за прокрутку вмісту?
21. Назвіть фундаментальну одиницю UI у React Native.
22. Як називається механізм, що передає дані від батьківського компонента до дочірнього?
23. Який об'єкт керує внутрішнім, змінюваним станом компонента?
24. Як називається функція-хук, що дозволяє використовувати внутрішній стан у функціональних компонентах?
25. Який цикл життя компонента відбувається при його первинному відображенні?
26. Яка функція компонента-класу обов'язково повинна бути реалізована для відображення UI?
27. Що використовується для створення інкапсульованих об'єктів стилів у React Native?
28. Який принцип верстки використовується для побудови макетів у React Native?
29. Як називається сучасний підхід до стилізації, що використовує JS для визначення стилів?
30. Який принцип стилізації фокусується на тому, щоб UI виглядав добре на різних розмірах екрана?
31. Що таке Custom Component?
32. Який хук замінює методи життєвого циклу для функціональних компонентів?
33. Як називається процес перетворення компонентів у нативні елементи UI?
34. Який тип позиціонування елементів за замовчуванням використовується у React Native?
35. Як називається компонент, що є контейнером для інших елементів?
36. Що використовується для відображення текстового вмісту?
37. Яка властивість використовується для активації гнучкого позиціонування елементів за однією віссю?
38. Як називається підхід, що використовує допоміжні класи для стилізації, схожі на відомий CSS-фреймворк?
39. Яка властивість стилю контролює прозорість елемента?

40. Що використовується для створення платформозалежного коду (наприклад, для iOS та Android) в одному файлі?
41. Яка бібліотека є галузевим стандартом для керування навігацією у React Native?
42. Який тип навігації використовується для переходу між екранами з використанням історії (як у браузері)?
43. Який тип навігації відображає вкладки в нижній частині екрана?
44. Як називається навігаційний елемент, що висувається збоку екрана?
45. Який об'єкт використовується для передачі даних між екранами під час навігації?
46. Що використовується для обробки складних дотиків і жестів (свайпи, щипки)?
47. Який компонент у React Native обробляє натискання?
48. Як називається модуль, що дозволяє створювати плавні переходи та рух елементів?
49. Як називається техніка в анімації, що дозволяє змінювати одне значення в анімації на інше?
50. Який метод використовується для запуску анімації в Animated API?
51. Як називається процес, що дозволяє анімації відбуватися плавно, незалежно від JS-потоків?
52. Що використовується для створення сенсорних інтерфейсів?
53. Яка функція-хук дозволяє відстежувати зміни в навігаційному стані?
54. Який елемент керує загальним навігаційним потоком всередині застосунку?
55. Як називається спеціалізований тип View, що реагує на жести?
56. Яка бібліотека для керування станом обрана основною у новій програмі?
57. Як називається сховище (контейнер) глобального стану в Zustand?
58. Як називаються функції в Zustand, що дозволяють ефективно витягувати частину стану?
59. Який інструмент у JavaScript використовується для управління асинхронним кодом, повертаючи успіх або помилку?
60. Яке ключове слово використовується для очікування завершення асинхронної операції?
61. Який об'єкт JavaScript дозволяє виконувати HTTP-запити?
62. Назвіть популярну бібліотеку для виконання HTTP-запитів, що часто використовується замість вбудованого fetch?
63. Як називається архітектурний стиль для організації взаємодії між клієнтом і сервером?
64. Який метод HTTP-запиту використовується для отримання даних?
65. Який метод HTTP-запиту використовується для створення нового ресурсу?
66. Який метод HTTP-запиту використовується для оновлення ресурсу?
67. Який механізм дозволяє зберігати дані на пристрої для офлайн-доступу?
68. Що таке локальне сховище?
69. Який стандарт використовується для безпечної автентифікації користувачів через зовнішні сервіси?
70. Що таке кешування даних?
71. Яка платформа, що базується на PostgreSQL, використовується як бекенд у новій програмі?
72. Як називається функція в Supabase, що забезпечує оновлення даних у реальному часі?
73. Яка платформа є альтернативою Firebase та Supabase, що часто позиціонується як Open Source?
74. Яка база даних лежить в основі Supabase?
75. Який тип операцій включає створення, читання, оновлення та видалення даних?
76. Який сервіс у Supabase/Appwrite відповідає за управління користувачами та їх сесіями?
77. Що таке Mobile Backend as a Service (MBaaS)?
78. Який тип тестування перевіряє невеликі ізольовані частини коду (функції, методи)?
79. Який тип тестування перевіряє взаємодію кількох модулів або компонентів?
80. Який тип тестування симулює дії кінцевого користувача всього застосунку?
81. Який фреймворк використовується для Unit-тестування у React Native?
82. Яка бібліотека допомагає тестувати React-компоненти, фокусуючись на взаємодії з користувачем?
83. Як називається процес виявлення та виправлення помилок у коді?
84. Що таке профілювання застосунку?
85. Який процес перетворює початковий код у готовий для встановлення файл?
86. Як називається інструмент, що допомагає збирати та публікувати Expo-проекти для магазинів?
87. Який магазин додатків використовується для публікації застосунків на платформі Android?
88. Який магазин додатків використовується для публікації застосунків на платформі iOS?
89. Як називається процес, що забезпечує швидке завантаження та плавну роботу застосунку?
90. Що таке збирання застосунку?
91. Який процес допомагає відстежувати продуктивність застосунку після його релізу?
92. Як називається процес випуску нової, покращеної версії застосунку?
93. Що таке функції (Functions) в Appwrite?
94. Як називається сховище файлів (наприклад, фото) у Supabase/Appwrite?
95. Який механізм дозволяє обійти проблеми з безпекою при автентифікації на зовнішніх сервісах?
96. Як називається компонент, який використовує функціонал, специфічний для однієї платформи?
97. Який тип навігації використовує єдиний потік екранів, накладаючи їх один на одного?
98. Як називається підхід до управління станом, що передбачає мінімалістичний та нешаблонний код?
99. Який механізм дозволяє застосунку реагувати на фонові події, надіслані сервером?
100. Як називається процес синхронізації даних між локальним сховищем і сервером?

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Мобільно-орієнтована розробка програмного забезпечення» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

1. Бобровнікова К.Ю., Медзатий Д.М., Павлова О.О. Мобільно-орієнтована розробка програмного забезпечення. Методичні вказівки для виконання лабораторних робіт. Хмельницький : ХНУ, 2020. - 134 с.

12. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Вивчення дисципліни «Мобільно-орієнтована розробка програмного забезпечення» вимагає використання сучасного програмного та технічного забезпечення:

Матеріально-технічне забезпечення:

- Аудиторне обладнання. Комп'ютерний клас з персональними комп'ютерами (ПК), мультимедійний проєктор.
- Персональне обладнання. ПК, ноутбук, планшет або смартфон для тестування застосунків на мобільній платформі.
- Мережа. Доступ до високошвидкісної мережі Інтернет.

Програмне забезпечення:

- **Операційні системи.** Windows, macOS або Linux.
- **Редактор коду.** Visual Studio Code або інше IDE з підтримкою TypeScript та React Native.
- **Середовище виконання.** Node.js та npm (або Yarn).
- **Кросплатформений фреймворк.** React Native.
- **Інструмент розробки.** Expo CLI (для швидкого старту та тестування).
- **Мова програмування.** TypeScript.
- **Керування станом.** Бібліотека Zustand.
- **Backend-сервіси.** Акаунти на платформах Supabase або Appwrite для роботи з даними та автентифікацією.
- **Система контролю версій.** Git.
- **Тестування.** Бібліотеки Jest та React Native Testing Library.
- **Навчальна платформа.** Доступ до Модульного середовища для навчання MOODLE.

13. Рекомендована література:

Основна література:

1. Masiello E., Friedmann J. Mastering React Native. Packt Publishing, 2020. 496 p. (Залишено як найновіше видання з наданого списку).
2. Flanagan D. JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides). O'Reilly Media, 2020. 1096 p. (Фундаментальний підручник з JS, що є основою React Native).
3. Офіційна документація React Native. Learn once, write anywhere. [Електронний ресурс]. Доступно: <https://reactnative.dev/> (Обов'язкове, найактуальніше джерело).
4. Бобровнікова К.Ю., Медзатий Д.М., Павлова О.О. Мобільно-орієнтована розробка програмного забезпечення. Методичні вказівки для виконання лабораторних робіт. Хмельницький: ХНУ, 2020. 134 с. (Обов'язкова навчально-методична праця авторів кафедри).
5. Simpson K. You Don't Know JS: Types & Grammar. O'Reilly Media, 2018. 200 p. (Включено як важливий ресурс для поглиблення знань JS/TypeScript, що є основою фреймворку).

Додаткова література та Інформаційні ресурси:

6. Eisenman B. Learning React Native: Building Native Mobile Apps with JavaScript. O'Reilly Media, Inc., 2019. 272 p.
7. Boduch A. React and React Native. Packt Publishing, 2019. 500 p.
8. Kho R. React Native By Example. Packt Publishing, 2019. 414 p.
9. Martinez E. R. React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript. Packt Publishing, 2019. 346 p.
10. Zakas N.C. High Performance JavaScript: Build Faster Web Application Interfaces. Yahoo Press, 2020. 232 p.
11. De Rosa A., Buckler C., Jacques N. et al. The Modern JavaScript Collection. SitePoint, 2018. 548 p.
12. Офіційна документація TypeScript. [Електронний ресурс]. Доступно: <https://www.typescriptlang.org/docs/>
13. Офіційна документація Expo. [Електронний ресурс]. Доступно: <https://docs.expo.dev/>
14. Документація по Zustand. [Електронний ресурс]. Доступно: <https://zustand.store/> (Ресурс для вивчення управління станом).
15. Документація по Supabase. [Електронний ресурс]. Доступно: <https://supabase.com/docs> (Ресурс для вивчення Backend-сервісу).
16. Документація по Appwrite. [Електронний ресурс]. Доступно: <https://appwrite.io/docs> (Ресурс для вивчення Backend-сервісу).

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=7519>
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. Після вивчення дисципліни студент повинен: знати основні концепції та поняття фреймворку React Native, зокрема компоненти, JSX, props, state, принципи створення нативних мобільних додатків, а також володіти базовими інструментами JavaScript/TypeScript для розробки мобільного програмного забезпечення. Студент повинен уміти застосовувати набуті знання для проєктування, реалізації та тестування мобільних додатків, обґрунтовувати вибір інструментів і технологій, таких як Zustand та Supabase/Appwrite, поєднувати теоретичні знання з практикою програмування, вирішувати прикладні завдання в умовах обмежених ресурсів. Також студент має демонструвати уміння працювати з документацією, проводити дослідження, оцінювати результати та аргументовано захищати технічні рішення, а також проявляти ініціативність, здатність до самостійного навчання, комунікації, роботи в команді та прийняття відповідальних інженерних рішень.

Зміст навчальної дисципліни. Застосування засобів мови JavaScript та фреймворку React Native для розробки нативних застосунків, зокрема для платформи Android. Навчання починається з вивчення відмінностей між React та React Native, аналізу переваг, ризиків і недоліків, а також налаштування середовища розробки в Integrated Development Environment. Основна увага приділяється компонентній моделі: створенню компонентів, рендерінгу, життєвому циклу, суті компонентного підходу, використанню JSX, вивченню базових та платформозалежних компонентів (для Android та iOS), стратегій їх включення, а також роботі з користувацькими компонентами, Props, State та Hook. Детально вивчаються питання стилізації: оголошення стилів, маніпулювання ними, методи позиціонування та побудова макетів. Курс охоплює роботу з API React Native (зокрема, специфічними API для платформи), встановлення та використання модулів сторонніх розробників, нативних і кросплатформених модулів. Значна частина присвячена взаємодії з користувачем, включаючи створення сенсорних інтерфейсів, оброблення дотиків і жестів, а також керуванню загальним навігаційним потоком (мобільна взаємодія та навігаційні шаблони). Розглядаються теми анімації, організації асинхронного коду через Promise, використання баз даних для мобільних застосунків (Firebase), управління станом програми за допомогою Redux, інтеграції RESTful API, Push Notifications та Facebook integration, а також відлагодження коду і тестування.

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: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); практичні та лабораторні заняття (з використанням пояснювально-ілюстративних, практичних та частково-пошукових методів); самостійна робота (передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів).

Форми оцінювання результатів навчання: оцінювання результатів роботи на практичних заняттях, оцінювання результатів захисту лабораторних робіт, усне опитування перед допуском до виконання лабораторної роботи та тестовий контроль.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Masiello E., Friedmann J. Mastering React Native. Packt Publishing, 2020. 496 p. (Залишено як найновіше видання з наданого списку).
2. Flanagan D. JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides). O'Reilly Media, 2020. 1096 p. (Фундаментальний підручник з JS, що є основою React Native).
3. Офіційна документація React Native. Learn once, write anywhere. [Електронний ресурс]. Доступно: <https://reactnative.dev/> (Обов'язкове, найактуальніше джерело).
4. Бобровнікова К.Ю., Медзатий Д.М., Павлова О.О. Мобільно-орієнтована розробка програмного забезпечення. Методичні вказівки для виконання лабораторних робіт. Хмельницький: ХНУ, 2020. 134 с. (Обов'язкова навчально-методична праця авторів кафедри).
5. Simpson K. You Don't Know JS: Types & Grammar. O'Reilly Media, 2018. 200 p. (Включено як важливий ресурс для поглиблення знань JS/TypeScript, що є основою фреймворку).
6. Eisenman B. Learning React Native: Building Native Mobile Apps with JavaScript. O'Reilly Media, Inc., 2019. 272 p.

7. Boduch A. React and React Native. Packt Publishing, 2019. 500 p.
8. Kho R. React Native By Example. Packt Publishing, 2019. 414 p.
9. Martinez E. R. React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript. Packt Publishing, 2019. 346 p.
10. Zakas N.C. High Performance JavaScript: Build Faster Web Application Interfaces. Yahoo Press, 2020. 232 p.
11. De Rosa A., Buckler C., Jacques N. et al. The Modern JavaScript Collection. SitePoint, 2018. 548 p.
12. Офіційна документація TypeScript. [Електронний ресурс]. Доступно: <https://www.typescriptlang.org/docs/>
13. Офіційна документація Expo. [Електронний ресурс]. Доступно: <https://docs.expo.dev/>
14. Документація по Zustand. [Електронний ресурс]. Доступно: <https://zustand.store/> (Ресурс для вивчення управління станом).
15. Документація по Supabase. [Електронний ресурс]. Доступно: <https://supabase.com/docs> (Ресурс для вивчення Backend-сервісу).
16. Документація по Appwrite. [Електронний ресурс]. Доступно: <https://appwrite.io/docs> (Ресурс для вивчення Backend-сервісу).
17. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=7519>
18. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
19. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

Викладачі: старший викладач Денисюк Д.О.