

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ




М. П. Савенко О.С.
2023 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мобільно-орієнтована розробка програмного забезпечення
Назва

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


Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин					Курсовий проєкт	Курсова робота	Форма семестрового контролю		
			Кредити ЕКТС	Години	Разом	Аудиторні заняття			Індивідуальна робота студента			Самостійна робота, в т.ч. ІРС	Залік	Іспит
						Лекції	Лабораторні роботи	Практичні заняття						
ОД		парний	8	240	102	17	68	17		138			+	

Програма складена  Нічепоруком А.О.
Підпис Ініціал, прізвище викладача(ів)

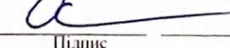
Схвалена на засіданні кафедри Комп'ютерної інженерії та інформаційних систем

Протокол № 1 від 30 06 2023 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем

 Говорушенко Т.О.
Підпис Ініціал, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради  Савенко О.С.
Підпис Ініціал, прізвище

ВСТУП

Мета викладання дисципліни. Дисципліна “Мобільно-орієнтована розробка програмного забезпечення” відноситься до циклу вибіркових дисциплін та забезпечує підготовку студентів з створення нативних мобільних застосунків з використанням фреймворку React Native.

Метою курсу є оволодіння студентами практичними навичками розроблення мобільних застосунків з використанням фреймворку React Native, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності.

Предмет дисципліни. Застосування засобів мови JavaScript та фреймворку React Native для розроблення нативних мобільних застосунків на прикладі платформи Android.

Завдання дисципліни:

- ознайомити з основними концепціями фреймворку React Native;
- ознайомити з засобами фреймворку React Native для створення нативних мобільних застосунків;
- навчити практичним навичкам розроблення мобільних застосунків з використанням засобів мови JavaScript та фреймворку React Native на прикладі платформи Android.

Після вивчення дисципліни «Мобільно-орієнтована розробка програмного забезпечення» студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

знати:

- теоретичні положення, що лежать в основі концепцій React Native; основні концепції фреймворку React Native, такі як компоненти, render, jsx, props, state; основні засоби JS, що використовуються в фреймворку React Native, та принципи програмування нативних мобільних застосунків;

вміти:

- здійснювати пошук, оброблення та аналіз інформації з різних джерел при розробленні мобільних застосунків;
- застосовувати знання основних концепцій фреймворку React Native при розв’язуванні задач аналізу та синтезу мобільних програмних систем;
- застосовувати знання теоретичних положень, що лежать в основі концепцій React Native, для ідентифікації, формулювання і розв’язування прикладних задач, використовуючи засоби фреймворку React Native, що є найбільш придатними для досягнення поставлених цілей;
- поєднувати теорію і практику розроблення мобільних застосунків, а також приймати оптимальні рішення при виробленні стратегії створення нативних мобільних застосунків з урахуванням виробничих інтересів;
- виконувати експериментальні дослідження з метою пошуку оптимальних шляхів при створенні мобільних застосунків з використанням концепцій фреймворку React Native;
- системно мислити та застосовувати творчі здібності до формування нових ідей при розробленні мобільних застосунків з використанням концепцій фреймворку React Native;
- оцінювати отримані результати та аргументовано захищати прийняті рішення при виборі засобів фреймворку React Native для розроблення мобільних застосунків;

бути здатним:

- до абстрактного мислення, аналізу і синтезу при розробленні мобільних застосунків з використанням концепцій фреймворку React Native;
- до навчання та оволодіння сучасними знаннями з метою вироблення стратегії створення мобільних застосунків з використанням концепцій фреймворку React Native та застосовувати одержані знання у практичних ситуаціях;
- до розуміння концепцій фреймворку React Native;
- застосовувати знання концепцій фреймворку React Native у практичних ситуаціях;

- використовувати поняття концепцій фреймворку React Native, а також принципи розроблення мобільних застосунків з використанням концепцій фреймворку React Native при розробленні мобільного програмного забезпечення на сучасних мовах програмування, зокрема JS;
- аргументувати вибір методів розроблення мобільних застосунків та засобів фреймворку React Native, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;
- розв'язувати складні задачі під час розроблення мобільних застосунків, що передбачає застосування концепцій фреймворку React Native і характеризується комплексністю та невизначеністю умов.

МОБІЛЬНО-ОРІЄНТОВАНА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	Парний
Кредити ЄКТС	8,0
Форми здобуття освіти	Очна денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: знати теоретичні положення, що лежать в основі концепцій React Native; основні концепції фреймворку React Native, такі як компоненти, render, jsx, props, state; основні засоби JS, що використовуються в фреймворку React Native, та принципи програмування нативних мобільних застосунків; вміло застосовувати знання теоретичних положень, що лежать в основі концепцій React Native, для ідентифікації, формулювання і розв'язування прикладних задач, використовуючи засоби фреймворку React Native, що є найбільш придатними для досягнення поставлених цілей; поєднувати теорію і практику розроблення мобільних застосунків, а також приймати оптимальні рішення при виробленні стратегії створення нативних мобільних застосунків; виконувати експериментальні дослідження з метою пошуку оптимальних шляхів при створенні мобільних застосунків з використанням концепцій фреймворку React Native; використовувати поняття концепцій фреймворку React Native, а також принципи розроблення мобільних застосунків з використанням концепцій фреймворку React Native при розробленні мобільного програмного забезпечення на сучасних мовах програмування, зокрема JS.

Зміст навчальної дисципліни. Застосування засобів мови JavaScript та фреймворку React Native для розробки нативних застосунків для платформи Android; React та React Native, їх відмінності; переваги для розробника; ризики і недоліки. Integrated Development Environment та особливості налаштування середовища розробки; довідники і специфікації. Створення компонентів, рендерінг, життєвий цикл компонента; суть компонентного підходу; використання JSX; базові компоненти React Native; платформозалежні компоненти; компоненти React Native для Android; Компоненти React Native для iOS; стратегії включення платформозалежних компонентів в крос-платформенні застосунки. Користувачькі компоненти. Props. State. Hook. Стилізація компонентів. Оголошення стилів та маніпулювання стилями. Методи позиціювання. Побудова макетів. API, API React Native для Android; API React Native для iOS. Специфічні API для платформи (доступ до телефону, камери, постійного сховища даних, геолокація). Модулі; встановлення та використання модулів сторонніх розробників; нативні модулі для Android; крос-платформенні модулі. Взаємодія з користувачем; створення сенсорних інтерфейсів (оброблення дотиків і жестів). Робота з організаційними компонентами (керування загальним навігаційним потоком всередині застосунку, реалізація мобільної взаємодії та навігаційних шаблонів). Анімація. Promise. Організація асинхронного коду. Бази даних для мобільних застосунків. Firebase. Redux. Управління станом програми. RESTful API. Push Notifications. Facebook integration. Відлагодження коду. Тестування.

Запланована навчальна діяльність: лекції – 17 год., лабораторні заняття – 68 год., практичні заняття – 17 год., самостійна робота – 138 год.; разом – 240 год.

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

Форми і методи оцінювання результатів навчання: захист лабораторних та практичних робіт, тестовий контроль.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. React Native 0.66. Learn once, write anywhere. Доступ до ресурсу: <https://reactnative.dev/>
2. Eisenman B. Learning React Native: Building Native Mobile Apps with JavaScript / B. Eisenman. – O'Reilly Media, Inc., 2019. – 272 p.
3. Boduch A. React and React Native / A. Boduch. – Packt Publishing, 2019. – 500 p.
4. Kho R. React Native By Example / R. Kho. – Packt Publishing, 2019. – 414 p.
5. Martinez E. R. React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript / E. R. Martinez. – Packt Publishing, 2019. – 346 p.
6. Masiello E. Mastering React Native / E. Masiello, J. Friedmann. – Packt Publishing, 2020. – 496 p.
7. De Rosa A. The Modern JavaScript Collection / A. De Rosa, C. Buckler, N. Jacques at all. – SitePoint, 2018. – 548 p.
8. Flanagan D. JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides) / D. Flanagan. – O'Reilly Media, 2020. – 1096 p.
9. Kolce J. JavaScript: Best Practice / J. Kolce, M. Kroger, I. Curic, S. Saeed at all. – SitePoint, 2018.
10. Kolce J. Modern JavaScript Tools & Skills / J. Kolce, M. Brown, C. Buckler at all. – SitePoint, 2018.
11. Zakas N.C. High Performance JavaScript: Build Faster Web Application Interfaces / N.C. Zakas. – Yahoo Press, 2020. – 232 p.
12. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua/course/view.php?id=7519>
13. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/plage_lib.php

Викладач: кандидат техн. наук, доцент Бобровнікова К.Ю.

1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Денна форма			
	Лекції	Лабораторні роботи	Практичні роботи	СРС
Тема 1. Введення в React Native.	4	8	2	16
Тема 2. Компоненти та API React Native.	4	28	10	48
Тема 3. Взаємодія з користувачем.	4	16	2	32
Тема 4. Бази даних для мобільних застосунків.	5	16	3	42
<i>Разом за семестр:</i>	17	68	17	138

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1.	Тема 1. Введення в React Native. Лекція № 1. Архітектура мобільного застосунку. Вимоги до компонентів. Архітектура сторінки. Масштабована архітектура для великих мобільних застосунків. Вимоги до великих проектів. Літ. [1, 3, 6]	2
2.	Тема 1. Введення в React Native. Лекція № 2. Введення в React Native. Переваги для розробника. Ризики і недоліки. React та React Native, їх відмінності. Синтаксичний цукор JSX. Integrated Development Environment. Довідники і специфікації. Особливості налаштування середовища розробки. Структура проекту. Основи JS. Літ. [1, 3-8]	2
3.	Тема 2. Компоненти та API React Native. Лекція № 3. Компоненти. Базові компоненти React Native. Платформозалежні компоненти. Компоненти React Native для Android. Компоненти React Native для iOS. Додаткові компоненти для специфічних застосунків. Створення компонентів. Рендерінг. Життєвий цикл компонента. Літ. [1-8]	2
4.	Тема 2. Компоненти та API React Native. Лекція № 4. Компонентний підхід. Стилїзування компонента. Поняття компонентного підходу. Стилїзація компонентів. Оголошення стилів та маніпулювання стилями. Конструкція StyleSheet.create. Прив'язка об'єктів стилів до компонентів. Методи позиціонування. Побудова макетів. Стратегії включення платформозалежних компонентів в крос-платформенні застосунки. Літ. [1-8]	2
5.	Тема 3. Взаємодія з користувачем. Лекція № 5. Організація асинхронного коду. Поняття Promise. Об'єкт Promise. Стани Promise. Використання. Успішне виконання. Передача інформації про помилку. Ланцюжки Promises. Приклади реалізації. Літ. [1, 2, 4]	2
6.	Тема 3. Взаємодія з користувачем. Лекція № 6. Redux. Управління станом програми. Основоположні принципи та переваги концепції Redux. Reducer. List component. Ініціалізація Redux. Огляд інтерактивних прикладів з пісочниці CodeSandbox. Літ. [1-4]	2
7.	Тема 4. Бази даних для мобільних застосунків. Лекція № 7. RESTful API. Push Notifications. Facebook integration. Поняття RESTful API. Архітектурні обмеження. Архітектурні елементи. Поняття Push Notifications. Впровадження в мобільні застосунки. Інтеграція Facebook у мобільний застосунок. Літ. [6-8]	2
8.	Тема 4. Бази даних для мобільних застосунків. Лекція № 8. Бази даних для мобільних застосунків. Принципи вибору бази даних. Зберігання даних. Розмір даних. Масштабування і швидкість. Моделювання даних. Протоколи безпеки даних. Сумісність. Синхронізація між БД та локальним сервером. Синхронізація між декількома рівнями даних. Оптимізація при втратах Інтернет-з'єднання. Масштабування. Оновлення застосунків та зміни БД. Літ. [6-8]	2

9.	Тема 4. Бази даних для мобільних застосунків. Лекція № 9. Програмування серверної частини. Мобільний бекенд як сервіс. Програмування серверної частини. Mobile Backend as a Service, MBaaS. Літ. [6-8]	1
	<i>Разом за семестр:</i>	17

2.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1.	<i>Лабораторна робота №1.</i> Налаштування середовища розробки для React Native. Створення проекту.	8
2.	<i>Лабораторна робота №2.</i> Компоненти React Native, їх властивості.	8
3.	<i>Лабораторна робота №3.</i> Стилізування компонентів. Компонентний підхід.	12
4.	<i>Лабораторна робота №4.</i> API-інтерфейси React Native.	8
5.	<i>Лабораторна робота №5.</i> Router. Керування навігаційним потоком всередині застосунку.	8
6.	<i>Лабораторна робота №6.</i> Залучення Cloud Firestore. Реєстрація та автентифікація користувача.	8
7.	<i>Лабораторна робота №7.</i> Залучення Cloud Firestore. Запис та читання даних.	8
8.	<i>Лабораторна робота №8.</i> Інтеграція Redux в React Native.	8
<i>Разом за семестр:</i>		68

2.3 Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
1.	<i>Практична робота №1.</i> Вибір та налаштування середовища розробки для React Native. Створення застосунку.	2
2.	<i>Практична робота №2.</i> Базові компоненти в React Native.	2
3.	<i>Практична робота №3.</i> Стилізування компонентів. Застосування компонентного підходу.	2
4.	<i>Практична робота №4.</i> API-інтерфейси в React Native.	2
5.	<i>Практична робота №5.</i> Створення користувацького компонента в React Native.	2
6.	<i>Практична робота №6.</i> Керування навігаційним потоком всередині застосунку.	2
7.	<i>Практична робота №7.</i> Робота з базами даних. Аутентифікація користувача.	2
8.	<i>Практична робота №8.</i> Робота з базами даних. Читання та запис даних.	3
<i>Разом за семестр:</i>		17

2.4 Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту практичних та лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань тощо.

Номер тижня	Вид самостійної роботи	К-ть годин
1.	Опрацювання лекційного матеріалу. Підготовка до ЛР1. Підготовка до захисту ЛР1.	16
2.	Опрацювання лекційного матеріалу. Підготовка до ЛР2. Підготовка до захисту ЛР2.	16
3.	Опрацювання лекційного матеріалу. Підготовка до ЛР3. Підготовка до захисту ЛР3.	16
4.	Опрацювання лекційного матеріалу. Підготовка до ЛР4. Підготовка до захисту ЛР4.	16
5.	Опрацювання лекційного матеріалу. Підготовка до ЛР5. Підготовка до захисту ЛР5.	16
6.	Опрацювання лекційного матеріалу. Підготовка до ЛР6. Підготовка до захисту ЛР6.	16
7.	Опрацювання лекційного матеріалу. Підготовка до ЛР7. Підготовка до захисту ЛР7.	16
8.	Опрацювання лекційного матеріалу. Підготовка до ЛР8. Підготовка до захисту ЛР8.	16
9.	Опрацювання лекційного матеріалу. Підготовка до тестування.	10
	<i>Разом за семестр:</i>	138

3. МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням словесних, наочних, проблемно-пошукових методів; практичні та лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

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

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

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо концепцій фреймворку React Native, здатна застосовувати базові засоби фреймворку React Native при розробленні мобільного програмного забезпечення.

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

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

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

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

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінка „зараховано” А виставляється студенту, який глибоко засвоїв методи розв'язування практичних задач та вміє їх раціонально застосувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „зараховано” В отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

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

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

неточності у відповіді.

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

Оцінка „незараховано” FХ виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань.

Як правило, оцінка „незараховано” F виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

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

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

Аудиторна робота		Самостійна, індивідуальна робота	Семестровий контроль (залік)
Лабораторні роботи №:	Практичні роботи №:	Тестовий контроль	
1-8	1-8	Т 1-4	за рейтингом
ВК:	0,7	0,1	0,2
			0

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–18	19-20
Оцінка	2	3	4	5

Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в онлайн режимі в модульному середовищі MOODLE. Викладач виставляє результати тестування згідно журналу оцінок модульного середовища MOODLE.

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

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

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Інституцій на шкала балів	Інституцій на оцінка	Критерії оцінювання	
A	4,75-5,00	5	Зараховано	
B	4,25-4,74	4		Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків.
C	3,75-4,24	4		Добре – повне знання навчального матеріалу з кількома незначними помилками.
D	3,25-3,74	3		Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками.
			Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією.	

E	3.00-3,24	3		Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	Незараховано	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2		Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Архітектура мобільного застосунку.
2. Вимоги до компонентів.
3. Архітектура сторінки.
4. Масштабована архітектура для великих мобільних застосунків.
5. Вимоги до великих проєктів.
6. Переваги для розробника React Native.
7. Ризики і недоліки React Native.
8. React та React Native, їх відмінності.
9. Синтаксичний цукор JSX.
10. Структура проєкту.
11. Особливості JS. Відмінності від інших мов.
12. Ajax, Promise.
13. Events.
14. Polyfills.
15. Базові компоненти React Native.
16. Платформозалежні компоненти.
17. Компоненти React Native для Android.
18. Компоненти React Native для iOS.
19. Додаткові компоненти для специфічних застосунків.
20. Створення компонентів.
21. Рендерінг.
22. Життєвий цикл компонента.
23. Користувацькі компоненти.
24. Props.
25. State.
26. Hook.
27. Компонент-функція.
28. Компонент-клас.
29. Платформозалежний код.
30. Поняття компонентного підходу.
31. Стилізація компонентів.
32. Оголошення стилів та маніпулювання стилями.
33. Конструкція StyleSheet.create.
34. Прив'язка об'єктів стилів до компонентів.
35. Методи позиціювання.
36. Побудова макетів.
37. Стратегії включення платформозалежних компонентів в крос-платформенні застосунки.
38. API React Native для Android.
39. API React Native для iOS.
40. Специфічні API для платформи (доступ до телефону, камери, постійного сховища даних, геолокація).
41. Нативні модулі для Android.
42. Нативні модулі для iOS.
43. Крос-платформенні модулі.
44. Встановлення та використання модулів сторонніх розробників.
45. Навігація.
46. Робота з організаційними компонентами.
47. Керування загальним навігаційним потоком всередині застосунку.
48. Реалізація навігаційних шаблонів.
49. Типи компонентів навігації.
50. React-native-router-flux.
51. React-native-navigation.
52. Animated API.
53. Конфігурування анімації.
54. Композиція анімацій.
55. Поєднання анімаційних значень.
56. Інтерполяція.

57. Відстеження динамічних значень в анімації.
58. Відстеження жестів при анімації.
59. Відповідь на поточне значення анімації.
60. Використання нативного драйвера.
61. LayoutAnimation API.
62. Створення сенсорних інтерфейсів.
63. Оброблення дотиків і жестів.
64. Поняття Promise. Об'єкт Promise.
65. Стани Promise. Використання.
66. Ланцюжки Promises.
67. Принципи вибору бази даних.
68. Firebase. Firebase Realtime. Можливості, переваги використання та обмеження.
69. Доступ до бази даних Firebase Realtime.
70. Формат JSON.
71. Встановлення, налаштування Firebase Realtime.
72. Читання / запис даних.
73. Реєстрація, авторизація.
74. Управління станом програми. Основні принципи та переваги концепції Redux.
75. Reducer. List component.
76. Ініціалізація Redux.
77. RESTful API.
78. Push Notifications.
79. Інтеграція Facebook у мобільний застосунок.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

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

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

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. React Native 0.66. Learn once, write anywhere. Доступ до ресурсу: <https://reactnative.dev/>
2. Eisenman B. Learning React Native: Building Native Mobile Apps with JavaScript / B. Eisenman. – O'Reilly Media, Inc., 2019. – 272 p.
3. Boduch A. React and React Native / A. Boduch. – Packt Publishing, 2019. – 500 p.
4. Kho R. React Native By Example / R. Kho. – Packt Publishing, 2019. – 414 p.
5. Martinez E. R. React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript / E. R. Martinez. – Packt Publishing, 2019. – 346 p.
6. Masiello E. Mastering React Native / E. Masiello, J. Friedmann. – Packt Publishing, 2020. – 496 p.
7. De Rosa A. The Modern JavaScript Collection / A. De Rosa, C. Buckler, N. Jacques at all. – SitePoint, 2018. – 548 p.
8. Flanagan D. JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides) / D. Flanagan. – O'Reilly Media, 2020. – 1096 p.
9. Kolce J. JavaScript: Best Practice / J. Kolce, M. Kroger, I. Curic, S. Saeed at all. – SitePoint, 2018.
10. Kolce J. Modern JavaScript Tools & Skills / J. Kolce, M. Brown, C. Buckler at all. – SitePoint, 2018.
11. Simpson K. You Don't Know JS: Async & Performance / K. Simpson. – O'Reilly Media, 2018. – 298 p.
12. Simpson K. You Don't Know JS: ES6 & Beyond / K. Simpson. – O'Reilly Media, 2018. – 280 p.
13. Simpson K. You Don't Know JS: Scope & Closures / K. Simpson. – O'Reilly Media, 2018. – 98 p.
14. Simpson K. You Don't Know JS: this & Object Prototypes / K. Simpson. – O'Reilly Media, 2018. – 176 p.
15. Simpson K. You Don't Know JS: Types & Grammar / K. Simpson. – O'Reilly Media, 2018. – 200 p.
16. Simpson K. You Don't Know JS: Up & Going / K. Simpson. – O'Reilly Media, 2018. – 90 p.
17. Zakas N.C. High Performance JavaScript: Build Faster Web Application Interfaces / N.C. Zakas . – Yahoo Press, 2020. – 232 p.

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету.