



Програмування інтерфейсів користувача

Робоча програма дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Другий (магістерський)
Галузь знань	12 Інформаційні технології
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	ОНП Системне програмування та спеціалізовані комп'ютерні системи
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)
Рік підготовки, семестр	1 курс, весняний семестр
Обсяг дисципліни	5 кредитів (150 годин)
Семестровий контроль/ контрольні заходи	РГР, екзамен
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор і лабораторні: доцент кафедри СП і СКС Петрашенко Андрій Васильович, petrashenko@gmail.com
Розміщення курсу	Google Classroom

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна «Програмування інтерфейсів користувача» спрямована на вивчення технологічних аспектів створення сучасних інтерфейсів користувача

Метою кредитного модуля є формування у студентів здатності створювати програмне забезпечення для формування Web та мобільних інтерфейсів користувача, зокрема, обирати інструментальні засоби та програмні бібліотеки відповідно до визначених вимог до Web-орієнтованої системи.

Предмет навчальної дисципліни – процес прототипування та розроблення сучасних інтерфейсів користувача.

Вивчення дисципліни «Програмування інтерфейсів користувача» сприяє формуванню у студентів наступних компетентностей.

Компетентності, формуванню яких сприяє дана дисципліна:

ЗК 2 Здатність вчитися і оволодівати сучасними знаннями

ЗК 3 Здатність застосовувати знання у практичних ситуаціях

ЗК 7 Здатність виявляти, ставити та вирішувати проблеми

ЗК 8 Здатність працювати в команді

ФК 2 Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення

ФК 11 Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів

ФК 12 Здатність ідентифікувати, класифікувати та описувати роботу програмнотехнічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонент шляхом використання аналітичних методів і методів моделювання.

ФК 15 Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення

ФК 16 Здатність до алгоритмічного та логічного мислення

Формування зазначених компетентностей сприяє досягненню наступних **програмних результатів навчання**:

ПРН 3 Знати новітні технології в галузі комп'ютерної інженерії.

ПРН 6 Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН 7 Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН 8 Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН 11 Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН 12 Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН 14 Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

ПРН 16 Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН 18 Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.

ПРН 19 Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

ПРН 20 Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

ПРН 21 Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни «Програмування інтерфейсів користувача» необхідно і достатньо мати навички алгоритмічного програмування і відповідних мов та інструментальних засобів, засвоїти дисципліни «Web-дизайн» та «Об'єктно-орієнтоване програмування», а також мати базовий рівень володіння англійською мовою не нижче B1.

Успішне оволодіння знаннями з даної дисципліни забезпечує вивчення надалі таких курсів, як «Технологія розробки Web-додатків» та «Технологія програмування».

3. Зміст навчальної дисципліни

РОЗДІЛ 1. Веб-програмування інтерфейсів користувача

Тема 1.1. Вступ до ReactJS Основні поняття та принципи роботи з ReactJS. Створення та налаштування проекту. Компоненти та JSX.

Тема 1.2. Стан та події у ReactJS Управління станом компонентів. Використання хуків (Hooks). Обробка подій та форми.

Тема 1.3. Маршрутизація у ReactJS Огляд бібліотеки React Router. Створення динамічних маршрутів. Управління навігацією та передача параметрів.

Тема 1.4. Менеджмент стану у ReactJS Контекст API. Redux та інші бібліотеки для управління станом. Сінхронізація стану між компонентами.

Тема 1.5. Оптимізація та продуктивність у ReactJS Профілювання та оптимізація компонентів. Використання мемоізації. Асинхронні операції та завантаження даних.

РОЗДІЛ 2. Мобільне програмування інтерфейсів користувача

Тема 2.1. Основи програмування під Android Архітектура Android. Створення проекту в Android Studio. Основні компоненти Android-застосунків (Activity, Fragment).

Тема 2.2. Розробка інтерфейсів для Android Створення користувацьких інтерфейсів за допомогою XML. Використання RecyclerView та інших компонентів UI. Адаптація інтерфейсу під різні розміри екранів.

Тема 2.3. Програмування на Flutter Вступ до Flutter. Створення кросплатформених мобільних додатків. Використання віджетів та управління станом.

Тема 2.4. Навігація та маршрутизація у Flutter Огляд навігаційної системи Flutter. Створення маршрутів та передача даних. Управління навігацією та стеком екрану.

Тема 2.5. Інтеграція з платформами у Flutter Використання платформ-каналів для взаємодії з нативними API. Інтеграція з сторонніми бібліотеками та сервісами.

РОЗДІЛ 3. Desktopне програмування інтерфейсів користувача

Тема 3.1. Вступ до ElectronJS Основи ElectronJS. Створення першого десктопного додатку. Використання HTML, CSS та JavaScript у ElectronJS.

Тема 3.2. Інтерфейси та модулі у ElectronJS Створення користувацьких інтерфейсів. Використання вбудованих модулів ElectronJS. Управління вікнами та меню.

Тема 3.3. Розширені можливості ElectronJS Інтеграція з нативними API операційної системи. Використання системних сповіщень та інтеграція з файловою системою.

Тема 3.4. Основи програмування з nw.js Огляд nw.js. Встановлення та налаштування проекту. Використання веб-технологій для створення десктопних застосунків.

Тема 3.5. Порівняння ElectronJS та nw.js Переваги та недоліки кожного підходу. Вибір технології для проекту. Приклади реальних застосунків.

РОЗДІЛ 4. Тестування інтерфейсів користувача

Тема 4.1. Тестування з Jest Основи юніт-тестування з Jest. Написання та виконання тестів. Мокінг та асинхронне тестування.

Тема 4.2. Інтеграційне тестування з Cypress Вступ до Cypress. Написання інтеграційних тестів. Тестування взаємодії компонентів та поведінки користувача.

Тема 4.3. Тестування мобільних додатків Інструменти для тестування Android та Flutter додатків. Використання Espresso та Flutter Test. Автоматизація тестів та CI/CD інтеграція.

Тема 4.4. Тестування десктопних додатків Інструменти для тестування ElectronJS та nw.js додатків. Використання Spectron та інших бібліотек. Автоматизація тестів та забезпечення якості.

РОЗДІЛ 5. Практика та впровадження інтерфейсів користувача

Тема 5.1. Розгортання веб-застосунків Процес розгортання веб-додатків. Використання CI/CD для автоматизації. Налаштування середовища продакшн.

Тема 5.2. Оптимізація продуктивності мобільних та десктопних застосунків Методи покращення продуктивності. Аналіз та моніторинг. Використання інструментів для оптимізації.

Тема 5.3. Забезпечення безпеки додатків Основи безпеки веб, мобільних та десктопних додатків. Захист від основних вразливостей. Впровадження безпечних практик програмування.

Тема 5.4. Взаємодія з бекендом Методи інтеграції з серверними API. Використання REST та GraphQL. Обробка запитів та управління даними.

Тема 5.5. Підтримка та оновлення додатків Практики підтримки живих проектів. Управління версіями та оновлення. Реагування на зворотній зв'язок від користувачів.

4. Навчальні матеріали та ресурси

Базова література

1. Alex Banks, Eve Porcello. *Learning React: Modern Patterns for Developing React Apps*. - O'Reilly Media – 2020, 310p.
2. Steven Kinney. *Electron in Action*. - Manning Publications – 2020, 344p.

Допоміжна література

3. Adam Boduch, John Armstrong. *React and React Native: A complete hands-on guide to modern web and mobile development with React.js*, 3rd Edition. - Packt Publishing – 2020, 526p.
4. Lucas Fernandes da Costa. *Testing JavaScript Applications*. - Manning Publications – 2021, 512p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальний контент дисципліни складається з лекційних та лабораторних занять.

5.1. Лекційні заняття:

№ з/п	Назва теми лекції
1	Вступ до ReactJS Основні поняття та принципи роботи з ReactJS. Створення та налаштування проекту. Компоненти та JSX.
2	Стан та події у ReactJS Управління станом компонентів. Використання хуків (Hooks). Обробка подій та форми.
3	Маршрутизація у ReactJS Огляд бібліотеки React Router. Створення динамічних маршрутів. Управління навігацією та передача параметрів.
4	Менеджмент стану у ReactJS Контекст API. Redux та інші бібліотеки для управління станом. Сінхронізація стану між компонентами.
5	Оптимізація та продуктивність у ReactJS Профілювання та оптимізація компонентів. Використання мемоізації. Асинхронні операції та завантаження даних.
6	Основи програмування під Android Архітектура Android. Створення проекту в Android Studio. Основні компоненти Android-застосунків (Activity, Fragment).
7	Розробка інтерфейсів для Android Створення користувацьких інтерфейсів за допомогою XML. Використання RecyclerView та інших компонентів UI. Адаптація інтерфейсу під різні розміри екранів.
8	Програмування на Flutter Вступ до Flutter. Створення кросплатформених мобільних додатків. Використання віджетів та управління станом.
9	Навігація та маршрутизація у Flutter Огляд навігаційної системи Flutter. Створення маршрутів та передача даних. Управління навігацією та стеком екрану.
10	Інтеграція з платформами у Flutter Використання платформ-каналів для взаємодії з нативними API. Інтеграція з сторонніми бібліотеками та сервісами.
11	Вступ до ElectronJS Основи ElectronJS. Створення першого десктопного додатку. Використання HTML, CSS та JavaScript у ElectronJS.
12	Інтерфейси та модулі у ElectronJS Створення користувацьких інтерфейсів. Використання вбудованих модулів ElectronJS. Управління вікнами та меню.
13	Розширені можливості ElectronJS Інтеграція з нативними API операційної системи. Використання системних сповіщень та інтеграція з файловою системою.
14	Основи програмування з nw.js Огляд nw.js. Встановлення та налаштування проекту. Використання веб-технологій для створення десктопних застосунків.

15	Тестування з Jest Основи юніт-тестування з Jest. Написання та виконання тестів. Мокінг та асинхронне тестування.
16	Інтеграційне тестування з Cypress Вступ до Cypress. Написання інтеграційних тестів. Тестування взаємодії компонентів та поведінки користувача.
17	Тестування мобільних додатків Інструменти для тестування Android та Flutter додатків. Використання Espresso та Flutter Test. Автоматизація тестів та CI/CD інтеграція.
18	Тестування десктопних додатків Інструменти для тестування ElectronJS та nw.js додатків. Використання Spectron та інших бібліотек. Автоматизація тестів та забезпечення якості.

5.2. Лабораторні роботи:

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	Розробка клієнтської частини Web-додатка	9
2	Тестування клієнтської частини Web-додатка	9

5.3. Розрахунково-графічна робота (РГР)

РГР передбачає виконання індивідуальних завдань згідно до встановлених варіантів. Метою РГР є поглиблення знань і напрацювання навичок створення мобільних додатків, орієнтованих на роботу з реляційними базами даних. Ваговий бал РГР – 20 балів.

В умовах дистанційного навчання 2024-2025 навчального року усі види занять, у тому числі контрольні заходи, проводяться з використанням сервісу Zoom.

6. Самостійна робота студента/аспіранта

Самостійна робота студентів полягає в наступному:

- підготовці до лекційних занять шляхом вивчення попереднього лекційного матеріалу а також літературних джерел, на яких базується матеріал попередніх лекцій (список джерел і перелік розділів надається разом з лекційним матеріалом);
- підготовці до лабораторних робіт шляхом з ознайомлення із завданням та методичними вказівками до виконання лабораторної роботи, в т. ч. вивченні теоретичного матеріалу, необхідного для відповіді на контрольні питання до лабораторних робіт;
- виконанні завдання лабораторних робіт і оформленні звіту, який висвітлює результати виконання лабораторної роботи (лістинг коду, аналіз результатів і т. і.).

Контроль знання на лабораторних заняттях здійснюється шляхом перевірки звіту з лабораторних робіт та опитування, а також через виконання модульних контрольних робіт.

Термін виконання лабораторних робіт визначається індивідуально для кожної лабораторної роботи і вказується в методичних вказівках.

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які ставляться перед студентом:

- студент зобов'язаний відвідувати лекційні та лабораторні заняття й активно працювати над засвоєнням матеріалу, що викладається на них;
- на лекції викладач користується власним презентаційним матеріалом; відпрацьовує практичну частину на віртуальній машині;
- лабораторні роботи захищаються на лабораторному занятті, попередньо оформивши звіт, і надіславши його викладачу;
- модульна контрольна робота пишеться на лекційному занятті із застосуванням всіх доступних матеріалів;
- штрафні бали нараховуються за: невчасну здачу лабораторної роботи. Граничний термін захисту лабораторної роботи без нарахування штрафних балів визначається для кожної лабораторної роботи і надається студенту разом із завданням. Кількість штрафних балів не більше 12.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Види контролю та система оцінювання:

- 2 лабораторні роботи. Кожна лабораторна робота має наступну систему оцінювання:
 - бездоганна робота – 20 балів;
 - є певні недоліки у оформленні роботи – 19-16 балів;
 - є певні недоліки у виконанні програми роботи – 15-10 балів;
 - за невчасне виконання і захист лабораторної роботи нараховуються штрафні бали (до 3-х балів);
 - замість будь-якої однієї ЛР дозволяється зарахувати пройдений студентом курс відповідної тематики на платформах Coursera, Udeму тощо (за наявності відповідного сертифіката) – 15-20 балів.
- 1 розрахунково-графічна робота (РГР), яка має наступну систему оцінювання:
 - бездоганна робота – 20 балів;
 - є певні недоліки у оформленні роботи – 19-16 балів;
 - є певні недоліки у виконанні програми роботи – 15-10 балів;
 - за невчасне виконання і захист лабораторної роботи нараховуються штрафні бали (до 2-х балів);
- 4 контрольні роботи (КР) — 10 балів. Контрольне завдання цієї роботи складається з тестових запитань. За кожну правильну відповідь по тестовому запитанню нараховується 1 бал;

Отриманий бал за лабораторні роботи, РГР та контрольні роботи (максимум 100) враховується з коефіцієнтом 0.7, що дає змогу отримати максимум 70 балів.

- семестровий контроль — екзамен. На екзамені студенти виконують контрольну роботу, що складається із 15 тестових завдань. За кожну правильну відповідь по тестовому запитанню нараховуються 2 бали. Максимальний бал за екзамен – 30 балів.

Календарний контроль (атестація): провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою першої атестації є отримання не менше 10 балів (на час атестації). Умовою другої атестації – отримання не менше 30 балів (на час атестації).

Умови допуску до семестрового контролю (екзамену) є зарахування усіх лабораторних робіт та РГР і семестровий рейтинг більший або рівний 42 балам.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
<i>100-95</i>	<i>Відмінно</i>
<i>94-85</i>	<i>Дуже добре</i>
<i>84-75</i>	<i>Добре</i>
<i>74-65</i>	<i>Задовільно</i>
<i>64-60</i>	<i>Достатньо</i>
<i>Менше 60</i>	<i>Незадовільно</i>
<i>Не виконані умови допуску</i>	<i>Не допущено</i>

9. Додаткова інформація з дисципліни (освітнього компонента)

В рамках дисципліни передбачено зарахування однієї лабораторної роботи у вигляді проходження онлайн-курсу на порталі Coursera або подібному.

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри СПіСКС, к.т.н., Петрашенком А.В.

Ухвалено кафедрою СПСКС (протокол № 6 від 03.01.2024р.)

Погоджено Методичною комісією факультету ПМ (протокол № 6 від 26.01.2024р.)