

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ЗВ'ЯЗКУ ім. О.С. ПОПОВА

Кафедра інформаційних технологій

“ЗАТВЕРДЖУЮ”
Зав.каф. інформаційних технологій

_____ 2020 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Скриптові мови програмування

Спеціальність: 122 Комп'ютерні науки
(шифр і назва спеціальності)

факультет, відділення **Інфокомунікацій та програмної інженерії**

Одеса
2020 рік

Робоча програма Скриптові мови програмування для студентів
за спеціальністю підготовки бакалавра 122 Комп'ютерні науки

Розробник програми: к.т.н., доц. Єгошина Г.А., ст.викл. Северин М.В.

Робоча програма затверджена на засіданні кафедри *Інформаційних технологій*

Протокол від “___” _____ 2019 року № __

Завідувач кафедри *Інформаційних технологій*

_____ (_____)

“___” _____ 20__ року

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 4	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Вибіркова
	Спеціальність підготовки <u>122 Комп'ютерні науки</u> (шифр і назва)	
Семестрів – 1	Спеціалізація (професійне спрямування): не передбачено	Рік підготовки:
Змістових модулів – 3		3-й
Індивідуальне науково-дослідне завдання _____ (назва)		Семестр
Загальна кількість годин 1-го семестру – 120		3.1
		Лекції
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 4	Освітньо-кваліфікаційний рівень: <i>бакалавр</i>	14 год.
		Практичні, семінарські
		14 год
		Лабораторні
		28 год.
		Самостійна робота
		64 год
		Індивідуальні завдання:
Не передбачено		
	Вид контролю: іспит	

2. Анотація дисципліни

Програма вивчення навчальної дисципліни «Скриптові мови програмування» складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності *122 Комп'ютерні науки*.

З появою HTML5 та CSS3 розробка клієнтських веб-додатків на мові JavaScript стала надзвичайно популярним сегментом. Розробники фреймворку React втілили майже революційний підхід в вирішенні питання перетворення браузера в одну з самих кращих платформ для розробки додатків та інтерфейсів. В React приділяється достатньо уваги тестуванню та високій якості коду, даний фреймворк сприяє використанню передового досвіду, що накоплений всією екосистемою JavaScript.

В курсі розглядаються особливості організації додатку, взаємодія з різними серверними технологіями, питання безпеки, продуктивності та розгортання додатків.

3. Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Скриптові мови програмування» є вивчення є формування компетентностей щодо використання сучасних технологій розроблення клієнтської частини веб-застосунків.

Дисципліна призначена для забезпечення чіткого уявлення загальних концепцій інтернет-програмування та використання сучасних засобів розробки клієнтської частини веб-додатків.

Завданням курсу є поглиблення теоретичних знань, що необхідні для формування базової уяви про архітектуру побудови веб-додатків на стороні фронтенда.

Завданнями вивчення навчальної дисципліни є:

- оволодіння термінологією та знаннями, що складають теоретичну основу проектування фронтенду веб-додатків;
- ознайомлення з принципами роботи з контентом та особливостями-об'єктно – орієнтованого програмування на JavaScript;
- ознайомлення з особливостями підключення фреймворку React;
- вивчення принципів компонентного підходу та особливостей роботи з методом `render()`;
- ознайомлення з методиками роботи з формами в React;
- проектування веб-додатків різного призначення за допомогою фреймворку React.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **компетентностей**:

загальних:

- здатність до абстрактного мислення, аналізу та синтезу
- здатність застосовувати знання у практичних ситуаціях;
- знання та розуміння предметної області та розуміння професійної діяльності;
- здатність вчитися й оволодівати сучасними знаннями;

- здатність до пошуку, оброблення та аналізу інформації з різних джерел
- здатність генерувати нові ідеї (креативність);
- здатність працювати в команді;
- здатність приймати обґрунтовані рішення;
- здатність оцінювати та забезпечувати якість виконуваних робіт.

фахових:

- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність розробляти корпоративні web-додатки із застосуванням сучасних технологій та інструментарію;
- здатність забезпечувати якість комп'ютерних систем та оцінювати їх показники якості з використанням відповідних моделей та засобів на всіх етапах розробки
- здатність застосовувати принципи, методи та алгоритми комп'ютерної графіки під час розробки графічних інтерфейсів взаємодії людини з комп'ютером.

Результати навчання даної дисципліни деталізують такі **програмні результати навчання:**

- Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук
- Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій

- Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук
- Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування
- Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт)
- Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем
- Розробляти програмне забезпечення систем різних архітектур з використанням відповідних сучасних технологій, патернів та сервісів, забезпечувати розподілені обчислення та розробляти документи та форми в web, описувати та розробляти web-сервіси
- Забезпечувати ефективне управління якістю комп'ютерних систем на основі використання відповідних методологій, моделей, та інструментальних засобів
- Володіти принципами, методами та алгоритмами комп'ютерної графіки, застосовувати їх у реалізації графічних та мультимедійних можливостей у прикладних системах
- Застосовувати знання концепцій та методів побудови крос-платформного програмного забезпечення, володіти декількома сучасними фреймворками, включно з крос-платформним графічним інтерфейсом користувача

На вивчення навчальної дисципліни відводиться 120 годин / 4 кредити ECTS.

4. Інформаційний обсяг навчальної дисципліни

Програма навчальної дисципліни складається з таких змістових модулів:

№ з/п	Назва теми
	Робота з контентом та об'єктно –орієнтоване програмування на JavaScript
1	Основи JavaScript. Умовні конструкції та цикли. Робота з масивами. Функції в JavaScript.
2	Робота this. Функція в функції. Прив'язка контенту. Методи call, apply, bind
3	Основи роботи з ООП на JavaScript. Успадкування на JavaScript.
	Фреймворк React. Основні поняття та принципи роботи
4	Підключення React. Використання const там let. Компонентний підхід. Основний компонент. Особливості методу render. Робота з атрибутами тегів та CSS.
5	Стан state та життєвий цикл. Події в React. Робота зі state та подіями. Робота з this. Зміна state. Умовний рендеринг. Змінні – елементи. Вбудовані умови. Запобігання рендерингу. State та цикли. Списки та ключі.
6	Робота з формами в React. Модифікація даних при виводі. Відправка форми. Робота з textarea, checkbox, select, radio
7	Робота з компонентами. Передача методу. Передача параметра з компоненту угору. Декілька компонентів. Видалення елементів

5. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усьо- го	у тому числі				
л		п	лаб	інд	с.р	
<i>1</i>	2	3	4	5	6	7
Змістовий модуль 1. Робота з контентом та об'єктно –орієнтоване програмування на JavaScript						
Основи JavaScript. Умовні конструкції та цикли. Робота з масивами. Функції в JavaScript.		2	2	4		
Робота this. Функція в функції. Прив'язка контенту. Методи call, apply, bind		2	2	4		
Основи роботи з ООП на JavaScript. Успадкування на JavaScript.		2	2	4		
<i>Разом за змістовним модулем 1</i>		2	2	4		20
Змістовий модуль 2. Фреймворк React. Основні поняття та принципи роботи						
Підключення React. Використання const та let. Компонентний підхід. Основний компонент. Особливості методу render. Робота з атрибутами тегів та CSS		2	2	2		
Стан state та життєвий цикл. Події в React. Робота зі state та подіями. Робота з this. Зміна state. Умовний рендеринг. Змінні – елементи. Вбудовані умови. Запобігання рендерингу. State та цикли. Списки та ключі.		2	2	2		
Робота з формами в React. Модифікація даних при виводі. Відправка форми. Робота з textarea, checkbox, select, radio		2	2	2		
Робота з компонентами. Передача методу. Передача параметра з компоненту угору. Декілька компонентів. Видалення елементів		2	2	2		
<i>Разом за змістовним модулем 2</i>		8	4	12		24
УСЬОГО	120	14	14	28		64

6. Темі практичних занять

№ з/п	Назва теми	Кількість годин
1	Основи синтаксису JavaScript. Функції в JavaScript.	2
2	Функції в JavaScript.	2
3	Основи роботи з ООП на JavaScript	2
4	Компонентний підхід. Робота з атрибутами тегів та CSS	2
5	Робота зі state та подіями. Умовний рендеринг.	2
6	Робота з формами в React.	2
7	Робота з компонентами.	2
8	Разом	14

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	JavaScript Basics. Частина 1	2
2	JavaScript Basics. Частина 2	2
3	Функції. Методи JavaScript для роботи з масивами	4
4	ООП в JavaScript. Геттери та сеттери. Успадкування класів в JavaScript	4
5	Робота з JSX: рендеринг, робота з атрибутами та CSS	4
6	Робота зі state та подіями в React	2
7	Робота з циклами в React	2
8	Робота з формами: input, submit, форми та масиви	2
9	Робота з формами: textarea, select, radio, checkbox	2
10	Редагування елементів. Використання масивів та умовних конструкцій	2
11	Робота з компонентами	2
12	Разом	28

8. Самостійна робота

№ з/п	Види роботи	Кількість годин
1	Проробка лекцій	14
2	Вивчення додаткового матеріалу до лекцій	10
3	Підготовка до практичних занять	20
4	Підготовка до лабораторних робіт	20
	Разом:	64

9. Методи навчання

Мультимедійні лекції, демонстрування, викладання-пояснення, практичні заняття та лабораторні роботи, метод проблемно-орієнтовного навчання, самостійне спостереження, запитання-бесіда, ілюстрування, виконання індивідуальних завдань.

10. Методи контролю

Контроль рівня виконавських умінь для перевірки практичної підготовки, усне опитування, опитування під час презентації та захисту індивідуальних завдань, іспит.

11. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота		Іспит	Сума
ЗМ1	ЗМ2		
20	50	30	100

Шкала оцінювання: національна та ECTS

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

12. Методичне, технічне й програмне забезпечення /обладнання

Презентації лекцій, комплекс навчально-методичного забезпечення дисципліни. В якості редактора для розробки веб-додатків можна використовувати будь-який безкоштовний продукт, наприклад, Notepad++ (notepad-plus-plus.org), Brackets (brackets.io), Eclipse (www.eclipse.org), Vim (www.vim.org), Visual Studio Code (code.visualstudio.com), Atom (atom.io), NetBeans (netbeans.org). В тому числі можна використовувати онлайн – середовища, наприклад, CodePen (<https://codepen.io/>), JSFiddle (<https://jsfiddle.net/>), CodeSandbox (<https://codesandbox.io>)

11. Рекомендована література

1. Вайк А. JavaScript в примерах / Вайк А. — К. : ДиаСофт, 2000. — 301 с.
2. Мэрдок К. Л. JavaScript: Наглядный курс создания динамических Web-страниц / Мэрдок К. Л. — М. : Диалектика, 2001. — 284 с
3. Коржинский С. Н. Настольная книга Web-мастера: эффективное применение HTML, CSS, и JavaScript / Коржинский С. Н. — М. : КноРус, 2000. — 300 с.
1. Дмитриева М.В. JavaScript. Экспрес курс / Дмитриева М.В. — СПб. : БХВ-Петербург, 2004. — 620 с.
2. Дронов В.А. JavaScript в Web-дизайне / Дронов В.А. — СПб. : БХВ-Петербург, 2002. — 370 с.
3. Рейсиг Дж. JavaScript. Профессиональные приемы программирования / Дж. Рейсиг. — СПб. : Питер, 2008. — 352 с.
4. Макфарланд Д. JavaScript и jQuery. Исчерпывающее руководство. 3-е издание / Д. Макфарланд // Эксмо, 2015 – 688 с.

12. Інформаційні ресурси

- <http://js-book.org.ua/>
- <http://nodeweekly.com/>
- <http://javascriptweekly.com/>
- <http://www.2ality.com/>
- <http://nolanlawson.com/>
- <http://www.w3.org/>
- <http://validator.w3.org/>
- <http://jigsaw.w3.org/css-validator/>
- <http://w3schools.com/>
- <https://www.codecademy.com/learn/javascript>
- http://bit.ly/lynda_js_training
- <https://teamtreehouse.com/learn/javascript>
- <https://itproger.com/course/react-js>
- <https://uk.reactjs.org/tutorial/tutorial.html>