

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ
Декан математичного факультету
С. І. Гоменюк
« 03 » _____ 09 2021 р.

АСИНХРОННЕ ПРОГРАМУВАННЯ У ВЕБ
РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

підготовки бакалавра
денної (очної) та заочної (дистанційної) форм здобуття освіти
спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Укладач: Тодоріко О.О. к.т.н., ст. викладач кафедри комп'ютерних наук

Обговорено та ухвалено
на засіданні кафедри комп'ютерних наук
Протокол № 2 від «01» вересня 2021 р.
Завідувач кафедри



С. В. Чопоров

Погоджено
з навчально-методичним відділом



О. В. Лещинська

Ухвалено науково-методичною радою
математичного факультету

Протокол № 1 від «02» вересня 2021 р.
Голова науково-методичної ради факультету



О. С. Пшенична

1. Опис навчальної дисципліни

1	2	3	
Галузь знань, спеціальність, освітня програма рівень вищої освіти	Нормативні показники для планування і розподілу дисципліни на змістові модулі	Характеристика навчальної дисципліни	
		очна (денна) форма здобуття освіти	заочна (дистанційна) форма здобуття освіти
Галузь знань 12 Інформаційні технології	Кількість кредитів – 5	Вибіркова	
		Цикл професійної підготовки спеціальності	
Спеціальність 126 Інформаційні системи та технології	Загальна кількість годин – 150	Семестр:	
		28 год.	8 год.
Освітньо-професійна програма Інформаційні системи та технології		Практичні	
		28 год.	8 год.
Рівень вищої освіти: бакалаврський	Кількість поточних контрольних заходів –18	Самостійна робота	
		94 год.	134 год.
		Вид підсумкового семестрового контролю: залік	

2. Мета та завдання навчальної дисципліни

Метою вивчення навчальної дисципліни «Асинхронне програмування у веб» є набуття здобувачами вищої освіти спеціальних знань та практичних навичок з питань створення додатків для глобальної мережі Інтернет з використанням асинхронного програмування для оптимізації високонавантажених програм із частим очікуванням системи.

Основними **завданнями** вивчення дисципліни «Асинхронне програмування у веб» є:

- засвоєння особливостей розробки веб-додатків;
- оволодіння загальними поняттями синхронного та асинхронного програмування;
- оволодіння знаннями з концепції та прикладів асинхронного програмування;
- засвоєння основних технологічних методів практичного застосування асинхронного програмування для розробки програмного продукту.

У результаті вивчення навчальної дисципліни студент повинен набути таких результатів навчання (знання, уміння тощо) та компетентностей:

Заплановані робочою програмою результати навчання та компетентності	Методи і контрольні заходи, що забезпечують досягнення результатів навчання та компетентностей
Результати навчання	
<p>Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.</p>	<p>Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення. Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.</p>
<p>Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх впровадження у професійній діяльності.</p>	<p>Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення. Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.</p>
<p>Використовувати системи віртуалізації та контейнеризації у розробці інформаційних систем.</p>	<p>Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення. Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.</p>
<p>Використовувати системи розподілу завдань, відстеження помилок і контролю версій програмного забезпечення у професійній діяльності.</p>	<p>Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення. Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.</p>
Компетентності	
<p>Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що</p>	<p>Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення.</p>

Заплановані робочою програмою результати навчання та компетентності	Методи і контрольні заходи, що забезпечують досягнення результатів навчання та компетентностей
характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій	Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.
Здатність застосовувати знання у практичних ситуаціях.	Методи навчання: пояснення, дискусія, виконання завдань лабораторних робіт. Методи контролю: опитування, захист лабораторних робіт, оцінювання звіту.
Здатність до розуміння предметної області та професійної діяльності.	Методи навчання: пояснення, дискусія, виконання завдань лабораторних робіт. Методи контролю: захист лабораторних робіт, опитування.
Здатність оцінювати та забезпечувати якість виконуваних робіт.	Методи навчання: лекційний метод, лекція-візуалізація, дискусія, виконання завдань лабораторних робіт, пояснення. Методи контролю: захист лабораторних робіт, опитування.
Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводжування інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.	Методи навчання: пояснення, дискусія, виконання завдань лабораторних робіт. Методи контролю: захист лабораторних робіт, опитування
Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).	Методи навчання: пояснення, дискусія, виконання завдань лабораторних робіт. Методи контролю: захист лабораторних робіт, опитування.

Міждисциплінарні зв'язки.

Дисципліна «Асинхронне програмування у веб» вимагає від студентів знань та умінь з дисциплін циклу загальної підготовки освітньої програми, а саме:

1. Web-програмування
2. Бази даних

3. Комп'ютерні мережі
4. Захист інформації
5. Технології розробки програмного забезпечення

3. Програма навчальної дисципліни

Змістовий модуль 1. Асинхронне програмування.

Розуміння асинхронного програмування. Синхронне програмування. Асинхронне програмування: концепція, реалізація, приклади. Тенденції. Запит даних. Інші рішення: конкурентність (concurrency), паралелізм (parallel execution) та багатопоточність (multithreading). Шаблони асинхронності.

Змістовий модуль 2. Веб програмування. WebSockets.

Веб програмування. Асинхронний веб. WebSockets. Принципи роботи HTTP, веб-сокетів. Браузер- веб-сервер. Задачі, які потребують використання веб-сокетів. Практичні приклади.

Змістовий модуль 3. Розуміння асинхронного програмування. JavaScript.

Розуміння синхронного та асинхронного JavaScript. JavaScript виконання синхронного та асинхронного коду. Функціональна модель роботи браузера. Call Stack, Web APIs, Event loop. Синхронні завдання. Асинхронні завдання.

Змістовий модуль 4. Async.js.

Синхронний та асинхронний JavaScript з Async/Await. Асинхронний потік керування Async.js. Специфікації, принципи роботи Async/Await. Колбеки (Callback). Проміси (Promise). Проміси та асинхронність. Цикли, контроль потоків, обмеження в Async/Await.

Змістовий модуль 5. Node.js.

Основи Node.js. Генерування подій у Node.js. Відмінності між платформою Node.js та браузером. V8 та інші JavaScript-движки. Асинхронне програмування у Node.js.

Змістовий модуль 6. PHP.

Асинхронний PHP. Чуйність програми. Паралелізм та асинхронність. PHP. Неблокуючий режим потоку. Використання сервера черг. ReactPHP, PHP-PM, swoole, RoadRunner, amphp, ext-async.

Змістовий модуль 7. Python.

Особливості асинхронного програмування в Python. Створення синхронного веб-сервера. Синхронне програмування Python. Різниця у поняттях паралелізму, concurrency, потоковості та асинхронності. Спільний паралелізм із блокуючими викликами. Кооперативний паралелізм із неблокуючими викликами Python.

Синхронні (блокуючі) HTTP дзвінки. Асинхронні (неблокуючі) HTTP виклики Python. Бібліотеки Python для асинхронного програмування: Asyncio, Tornado та Gevent.

Змістовий модуль 8. Асинхронність Python.

Асинхронність з простими функціями. Подієвий цикл. Асинхронність на колбеках. Генератори та подієвий цикл Round Robin. Асинхронність на генераторах. Корутини та yield from. Asyncio, async/await.

4. Структура навчальної дисципліни

Змістовий модуль	Усього годин	Аудиторні (контактні) години					Самостійна робота, год		Система накопичення балів		
		Усього годин	Лекційні заняття, год		Лабораторні, год		о/д ф.	з/дист ф.	Теор. зав-ня, к-ть балів	Лаб. зав-ня, к-ть балів	Усього балів
			о/д ф.	з/дист ф.	о/д ф.	з/дист ф.					
1	2	3	4	5	6	7	8	9	10	11	12
1	15	5	2	1	3	1	10	13	2	5	7
2	15	5	2	1	3	1	10	13	2	5	7
3	15	7	4	1	3	1	8	13	3	5	8
4	15	7	4	1	3	1	8	13	3	5	8
5	15	8	4	1	4	1	7	13	2	5	7
6	15	8	4	1	4	1	7	13	2	5	7
7	15	8	4	1	4	1	7	13	3	5	8
8	15	8	4	1	4	1	7	13	3	5	8
Усього за змістові модулі	120	56	28	8	28	8	64	104	20	40	60
Підсумковий семестровий контроль залік	30						30				40
Загалом		150						100			

5. Теми лекційних занять

№ змістового модуля	Назва теми	Кількість годин	
		о/д ф.	з/дист ф.
1	2	3	4
1	Асинхронне програмування.	2	1
2	Веб програмування. WebSockets	2	1
3	Розуміння асинхронного програмування. JavaScript.	4	1
4	Async.js.	4	1
5	Node.js.	4	1
6	PHP.	4	1
7	Python.	4	1
8	Асинхронність Python.	4	1
Разом		28	8

6. Теми практичних занять

№ змістового модуля	Назва теми	Кількість годин	
		о/д ф.	з/дист ф.
1	2	3	4
1	Лабораторна робота 1. Використання AJAX	3	1
2	Лабораторна робота 2. Веб-сокети.	3	1
3	Лабораторна робота 3. Синхронний та асинхронний JavaScript.	3	1
4	Лабораторна робота 4. Async/Await.	3	1
5	Лабораторна робота 5. Node.js	4	1
6	Лабораторна робота 6. Асинхронний PHP.	4	1
7	Лабораторна робота 7. Створення веб-сервера на Python.	4	1
8	Лабораторна робота 8. Asyncio.	4	1
Разом		28	8

7. Види і зміст поточних контрольних заходів *

№ змістового модуля	Вид поточного контрольного заходу	Зміст поточного контрольного заходу	Критерії оцінювання	Усього балів
1	2	3	4	5
1	Лабораторна робота 1. Використання AJAX	Завдання: Створити файл index.html, який містить JavaScript та два посилання, при натисканні на які на поточну сторінку асинхронно завантажується вміст різних файлів з даними в форматі JSON.	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитування	Орієнтовні питання: синхронне, асинхронне програмування, концепція, реалізація, приклади асинхронного програмування. Запит даних, шаблони асинхронності, технологія AJAX, формат JSON.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
2	Лабораторна робота 2. Веб-сокети.	Завдання: за допомогою веб-сокетів розробити додаток в режимі реального часу, наприклад, простий чат.	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитування	Орієнтовні питання: асинхронний веб, принципи роботи HTTP, веб-сокетів, браузер- веб-сервер.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
3	Лабораторна робота 3. Синхронні та асинхронні JavaScript.	Завдання: використовуючи синхронні та асинхронні функції написати зворотні виклики, які необхідно виконати при обробці певних завдань, таких як читання файлу, підключення до бази даних. Порівняти результат виконання.	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитування	Орієнтовні питання: синхронність та асинхронність JavaScript, функціональна модель роботи браузера. Call Stack, Web APIs, Event loop.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
4	Лабораторна робота 4. Async/Await.	Завдання: за допомогою колбэків та промісів написати виведення на екран анімованого круга з повідомленням, коли анімація буде завершена. Порівняти результати.	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитування	Орієнтовні питання: Async/Await, специфікації, принципи роботи,	Правильні відповіді – 1 бал; неповні відповіді – 0,5	1

		колбеки (Callback), проміси (Promise), цикли, контроль потоків, обмеження в Async/Await.	балів	
	Контрольн а робота №1	10 питань з переліку питань	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	10
5	Лаборатор на робота 5. Node.js	Завдання: за допомогою Node.js розробити асинхронний чат. Результат порівняти з чатом, розробленим за допомогою веб-сокетів (лр2).	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитува ння	Орієнтовні питання: Node.js, генерування подій, асинхронне програмування у Node.js.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
6	Лаборатор на робота 6. Асинхронн ий PHP.	Завдання: використовуючи асинхронні можливості PHP розробити асинхронний чат. Результат порівняти з чатами, розробленим за допомогою веб-сокетів та Node.js (лр2, 5).	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитува ння	Орієнтовні питання: асинхронний PHP, чуйність програми, паралелізм та асинхронність, неблокуючий режим потоку, використання сервера черг, ReactPHP, PHP-PM, swoole, RoadRunner, amphp, ext-async.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
7	Лаборатор на робота 7. Створення веб-сервера на Python	Завдання: використовуючи reverse engineering проаналізувати особливості створення синхронного веб-сервера, використання синхронних (блокуючих) та асинхронних (неблокуючих) HTTP викликів Python.	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитува ння	Орієнтовні питання: особливості синхронного та асинхронного програмування в Python, синхронний веб-сервер, різниця у поняттях паралелізму, concurrency, потоковості та асинхронності, спільний паралелізм із блокуючими викликами, кооперативний паралелізм із неблокуючими викликами Python; синхронні (блокуючі) та асинхронні (неблокуючі) HTTP виклики Python, бібліотеки Python для асинхронного програмування: Asyncio, Tornado та Gevent.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1

8	Лабораторна робота 8. Asyncio	Завдання: використовуючи Asyncio розробити асинхронний чат. Результат порівняти з чатами, розробленим за допомогою веб-сокетів, Node.js та PHP (лр2, 5, 6).	Повне виконання завдання оцінюється в 4 бали, за наявності незначних помилок у 3 бали, при наявності грубих помилок – 1-2 бали; відсутність результату – 0 балів	4
	Опитування	Орієнтовні питання: асинхронність з простими функціями, подієвий цикл, асинхронність на колбеках, генератори та подієвий цикл Round Robin, асинхронність на генераторах, корутини та yield from, Asyncio, async/await.	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	1
	Контрольна робота №2	10 питань з переліку питань	Правильні відповіді – 1 бал; неповні відповіді – 0,5 балів	10
Усього				60

8. Підсумковий семестровий контроль

Форма	Види підсумкових контрольних заходів	Зміст підсумкового контрольного заходу	Критерії оцінювання	Усього балів
1	2	3	4	5
Залік	Теоретичне завдання	Питання для підготовки: https://moodle.znu.edu.ua/course/view.php?id=13857	Контрольне опитування. Оцінюються правильність та повнота відповідей	20
	Практичне завдання	Індивідуальне завдання: https://moodle.znu.edu.ua/course/view.php?id=13857	Контрольне опитування. Оцінюються правильність та повнота відповідей	20
Усього за підсумковий семестровий контроль				40

9. Рекомендована література

Основна:

1. Marijn Haverbeke Eloquent JavaScript. 3rd edition: A Modern Introduction to Programming. No Starch Press, 2018.-448 p.
2. Herron David Node.js Web Development: Server-side web development made easy with Node 14 using practical examples, 5th Edition. Birmingham, United Kingdom: Packt Publishing Limited, 2020. 760p.
3. Bruce Dou Mastering Swoole PHP: Build High Performance Concurrent System with Async and Coroutines. Transfon Ltd, 2020.-334 p.

4. Hattingh, C. Using Asyncio in Python: Understanding Python's Asynchronous Programming Features. O'Reilly Media, Incorporated, 2020. 152 p. URL: <http://onreader.mdl.ru/UsingAsyncioPython3/content/index.html>

Додаткова:

1. Simpson Kyle. You don't Know JS. Async & Performance. O'Reilly Media, 2015. 296 p.

2. Cantelon Mike, Harter Marc, Holowaychuk T.J., Rajlich Nathan Node.js in Action. - Dreamtech Press India, 2014. 416 p.

3. Quan Nguyen Mastering Concurrency in Python: Create faster programs using concurrency, asynchronous, multithreading, and parallel programming Paperback: Packt Publishing, 2019. 446 p. <http://onreader.mdl.ru/MasteringConcurrencyInPython/content/index.html>

4. Zacccone, G. Python Parallel Programming Cookbook - Second Edition/ Giancarlo Zacccone. PackT Publishing, 2019. –370 p.

Інформаційні ресурси:

1. Moodle сторінка дисципліни URL: <https://moodle.znu.edu.ua/course/view.php?id=13857>

2. JavaScript Довідка URL: <http://яваскрипт.укр/>

3. JavaScript reference. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

4. Promises, async/await. URL:<https://javascript.info/async>

5. Modern Asynchronous JavaScript with Async and Await. URL: <https://nodejs.dev/learn/modern-asynchronous-javascript-with-async-and-await>

6. Async Await in Node.js – How to Master it? URL: <https://blog.risingstack.com/mastering-async-await-in-nodejs/>

7. PHP Manual. URL: <https://www.php.net/manual/en/getting-started.php>

8. Asynchronous PHP — Multiprocessing, Multithreading & Coroutines. URL: <https://divinglaravel.com/asynchronous-php>

9. Solomon Brad. Async IO in Python: A Complete Walkthrough. URL: <https://realpython.com/async-io-python/>

10. Asyncio-Asynchronous I/O. Coroutines and Tasks. URL: <https://docs.python.org/3/library/asyncio-task.html>