

**МЕЛІТОПОЛЬСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

ФАКУЛЬТЕТ ІНФОРМАТИКИ, МАТЕМАТИКИ ТА ЕКОНОМІКИ

Кафедра інформатики і кібернетики

Назва освітнього компонента <i>Нормативний/вибірковий</i>	Теорія програмування <i>Нормативний</i>
Ступінь освіти	Бакалавр
Освітня програма	Професійна освіта. Комп'ютерні технології
Рік викладання / Семестр / Курс (рік навчання)	2022 – 2023 / VII семестр / IV курс
Викладач	Конюхов Сергій Леонідович
Профайл викладача	http://inf.mdpu.org.ua/2017/10/25/konjuhov-sergij-leonidovich/
Контактний тел.	+38 050 9293065
E-mail	konukhov@mdpu.org.ua
Сторінка курсу в ЦОДТ МДПУ ім. Б.Хмельницького	https://dfn.mdpu.org.ua/course/view.php?id=132
Консультації	Онлайн-консультації: згідно графіку роботи кафедри інформатики і кібернетики через систему ЦОДТ МДПУ імені Богдана Хмельницького.

1. АНОТАЦІЯ

Навчальна дисципліна «Теорія програмування» з циклу нормативних компонентів здобувачів вищої освіти за освітньою програмою «Професійна освіта. Комп'ютерні технології» передбачає формування у здобувачів поглиблених знань, умінь і навичок в методології організації архітектури програмних систем.

Основною вивчення дисципліни є надання здобувачам систематизованих відомостей про основні прийоми програмування з урахуванням сучасних концепцій і тенденцій розвитку технологій програмування в середовищі Node.js, навчитися розв'язувати реальні науково-технічні задачі різної складності за допомогою комп'ютерів.

Здобувачі мають можливість ознайомитись з методологією розробки алгоритмів прикладних задач, кодування на платформі Node.js, налагодження додатків, оцінки достовірності отриманих результатів.

2. МЕТА ТА ЗАВДАННЯ ОСВІТНЬОГО КОМПОНЕНТА

Основна мета засвоєння навчальної дисципліни полягає в ознайомленні студентів з методологією організації архітектури програмних систем, ефективного розгортання інформаційних систем в середовищі Node.js; формування в них здатностей до самостійного проектування та розробки веб-сервісів і веб-додатків на платформі Node.js; закріплення теоретичних знань із структурного та об'єктно-орієнтованого програмування.

Завданнями навчальної дисципліни є одержання студентами теоретичних знань та практичних навиків щодо архітектури інформаційних систем та розробки веб-сервісів.

3. ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ, ЯКІ НАБУВАЮТЬСЯ ПІД ЧАС ОПАНУВАННЯ ОСВІТНІМ КОМПОНЕНТОМ

Загальні компетентності

ЗК 1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 2. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні (фахові) компетентності

ФК 1. Здатність використовувати у професійній діяльності основні положення, методи, принципи фундаментальних та прикладних наук.

ФК 15. Здатність до математичного та логічного мислення, формулювання та досліджування математичних моделей, зокрема дискретних, обґрунтованого вибору методів розв'язування теоретичних і прикладних задач в галузі комп'ютерних технологій.

ФК 18. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних методів та алгоритмів обчислень, структур даних, парадигм програмування на основі побудови відповідних моделей.

4. РЕЗУЛЬТАТИ НАВЧАННЯ

ПРН 2. Уміння самостійно здобувати систематичні знання в галузях професійної освіти та комп'ютерних технологій, осмислювати і робити обґрунтовані висновки на основі аналізу літературних джерел та експериментальних даних.

ПРН 17. Знання фундаментальних основ функціонування інформаційних систем та комп'ютерної техніки.

ПРН 20. Знання базових понять теорії алгоритмів, формальних моделей алгоритмів, питань обчислюваності, розв'язності та нерозв'язності масових проблем, понять складності алгоритмів.

ПРН 21. Знання структур даних та фундаментальних алгоритмів, методології та інструментальних засобів об'єктно-орієнтованого аналізу та проектування, особливостей різних парадигм програмування, принципів, моделей, методів і технологій проектування і розроблення програмних продуктів різного призначення.

ПРН 25. Знання принципів, інструментальних засобів, мов програмування та технологій створення веб-ресурсів, баз даних, розподілених застосувань, інтелектуальних інформаційних систем тощо.

ПРН 26. Уміння використовувати методи, технології та інструментальні засоби для проектування і розробки веб-застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти запити до них, створювати розподілені бази даних, інтелектуальні інформаційні системи.

ПРН 33. Знати і розуміти роль і місце фізико-математичних та комп'ютерних наук у загальній системі знань про природу та суспільство, у розвитку техніки й технологій та у формуванні світогляду особистості.

5. ОБСЯГ ОСВІТНЬОГО КОМПОНЕНТА

Вид заняття	Лекції	Лабораторні заняття	Самостійна робота
Кількість годин	16 годин	28 годин	46 годин

6. ПОЛІТИКА

Політика академічної поведінки та етики:

- не пропускати та не запізнюватися на заняття за розкладом;
- вчасно виконувати завдання лабораторних занять та самостійної роботи;
- на заняттях дотримуватись правил роботи у групі, шанобливо ставитись до поглядів один одного;
- вчасно та самостійно виконувати контрольні завдання;
- дотримуватись політики доброчесності під час виконання завдань;
- активно брати участь у побудові і реалізації індивідуальної освітньої траєкторії.

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: дотримуватись Кодексу академічної доброчесності МДПУ імені Богдана Хмельницького https://mdpu.org.ua/wp-content/uploads/2020/11/Kodeks-akadem-dobrochesnosti_2020.pdf та Положення про академічну доброчесність https://mdpu.org.ua/wp-content/uploads/2020/11/akademichna-dobrochesnist_2020.pdf.

Політика щодо відвідування: відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

7. СТРУКТУРА ОСВІТНЬОГО КОМПОНЕНТА

7.1. СТРУКТУРА ОСВІТНЬОГО КОМПОНЕНТА (ЗАГАЛЬНА)

Кількість годин	Тема	Форма діяльності (заняття, кількість годин)	Література	Завдання	Вага оцінки	Термін виконання
Блок 1. Вступ до Node.js						
15	Тема 1. Основи роботи з Node.js	Лекція (4 год.), практична робота (4 год.), самостійна робота (7 год.)	Основна : 1-4, Додаткова: 1-3	Виконання індивідуальних завдань	5	впродовж сьомого навчального семестру (перший періодичний контроль)
5	Тема 2. Сервер	Самостійна робота (5 год.)	Основна : 1-4, Додаткова: 1-3	Виконання завдань самостійної роботи	5	впродовж сьомого навчального семестру (перший періодичний контроль)

25	Тема 3. Фреймворк Express	Лекція (6 год.), практична робота (12 год.), самостійна робота (7 год.)	Основна : 1-4, Додаткова: 1-3	Виконання індивідуального завдання	5	впродовж сьомого навчального семестру (перший періодичний контроль)
Блок 2. Робота з базами даних						
15	Тема 4. Node.js і СКБД MySQL	Лекція (2 год.), практична робота (4 год.), самостійна робота (9 год.)	Основна : 1-4, Додаткова: 1-3	Виконання індивідуального завдання	5	впродовж сьомого навчального семестру (другий періодичний контроль)
15	Тема 5. Node.js і СКБД MongoDB	Лекція (2 год.), практична робота (4 год.), самостійна робота (9 год.)	Основна : 1-4, Додаткова: 1-3	Виконання індивідуального завдання	5	впродовж сьомого навчального семестру (другий періодичний контроль)
15	Тема 6. ORM-бібліотека Sequelize	Лекція (2 год.), практична робота (4 год.), самостійна робота (9 год.)	Основна : 1-4, Додаткова: 1-3	Виконання індивідуального завдання	5	впродовж сьомого навчального семестру (другий періодичний контроль)

7.2. СТРУКТУРА ОСВІТНЬОГО КОМПОНЕНТА (ЛЕКЦІЙНИЙ БЛОК)

Теми лекцій та питання, що вивчаються
<p>Тема 1. Основи роботи з Node.js Призначення Node.js. Встановлення. Створення першої програми. Модулі. Визначення конструкторів і об'єктів у модулі. Структура модулів. Об'єкт global і глобальні змінні. Пакетний менеджер NPM. Файл Package.json. Встановлення модулів. Додавання багатьох пактів. Видалення пакетів. Визначення команд. Інструмент nodemon.</p>
<p>Тема 2. Нові можливості Node.js Асинхронність в Node.js. Робота з файлами. Читання з файлу. Запис файлу. Видалення файлу. Події. Клас EventEmitter та спадкування від нього. Передача параметрів події. Потік даних Stream. Канал Pipe. Архівування.</p>
<p>Тема 3. Фреймворк Express Призначення. Конвеєр обробки запиту і middleware. Відправка відповіді. Статичні файли. Маршрутизація. Переадресація. Передача даних застосуванню. POST-запити і відправка форм. Параметри маршруту.</p>
<p>Тема 4. Рушій представлень Handlebars Представлення і рушій представлень Handlebars. Підключення. Умовна і циклічна генерація. Часткові представлення в Handlebars. Layout в Handlebars. Хелпери в Handlebars.</p>
<p>Тема 5. Рушії шаблонів EJS і PUG. Рушій EJS. Призначення. Встановлення. Часткові представлення. Рушій PUG. Часткові представлення і Layout в PUG. Створення API.</p>

Тема 6. Node.js і СКБД MySQL

Підключення до MySQL. Установка та закриття підключення. Виконання запитів до MySQL. Виконання запитів через execute(). Параметризація запитів. Promise API. Основні операції з даними.

Тема 7. Node.js і СКБД MongoDB

Призначення MongoDB. База даних. Підключення до бази даних. Колекції. Додавання даних в MongoDB. Отримання даних в MongoDB. Видалення документів в MongoDB. Оновлення документів в MongoDB. Express і MongoDB.

Тема 8. ORM-бібліотека Sequelize

Призначення Sequelize. Підключення до баз даних. Визначення моделей. Запити до бази даних. Express і Sequelize.

7.3. СТРУКТУРА ОСВІТНЬОГО КОМПОНЕНТА (ЛАБОРАТОРНІ ЗАНЯТТЯ)

Теми лабораторних занять
Тема: Перший сайт на Node.JS <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «Основи роботи з Node.js»2. Виконання практичних завдань
Тема: Перший додаток на Express <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «Фреймворк Express»2. Виконання практичних завдань
Тема: Текстові файли <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «Фреймворк Express»2. Виконання практичних завдань
Тема: Використання статичного контенту <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «Фреймворк Express»2. Виконання практичних завдань
Тема: Бази даних <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «Node.js і СКБД MySQL»2. Виконання практичних завдань
Тема: Шаблони <ol style="list-style-type: none">1. Актуалізація теоретичних знань з теми «ORM-бібліотека Sequelize»2. Виконання практичних завдань

7.4. СТРУКТУРА ОСВІТНЬОГО КОМПОНЕНТА (ТЕМИ ДЛЯ САМОСТІЙНОГО ОПРАЦЮВАННЯ)

Тема для самостійного опрацювання
Тема: Основи роботи з Node.js Питання: Засоби розробки. Передача параметрів застосуванню. Семантичне версіонування.
Тема: Сервер Питання: Створення сервера. Параметри Request і Response. Маршрутизація. Переадресація. Відправка файлів. Отримання даних від клієнта. Шаблони.

Тема: Фреймворк Express
Питання: JSON і AJAX.
Тема: Node.js і СКБД MySQL
Питання: Пули підключень. Express і MySQL.
Тема: Node.js і СКБД MongoDB
Питання: Бібліотека Mongoose. Визначення схеми в Mongoose. CRUD в Mongoose. Express і Mongoose.
Тема: ORM-бібліотека Sequelize
Питання: Зв'язок один-до-багатьох. Зв'язок один-до-одного. Зв'язок багато-до-багатьох.

8. МЕТОДИ ТА ФОРМИ КОНТРОЛЮ

Поточний контроль: усне опитування, перевірка виконання практичних завдань. Періодичний контроль: письмова контрольна робота з виконанням практичного завдання; презентація навчального проекту. Підсумковий семестровий контроль.

КРИТЕРІЇ ОЦІНЮВАННЯ ВІДПОВІДНО ДО ВИДІВ КОНТРОЛЮ

Критерії оцінювання діяльності студентів на лабораторних заняттях, виконання завдань самостійного опрацювання наведені у таблиці:

Бали	Критерії
5	Студент в повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час усних виступів та письмових відповідей, глибоко та всебічно розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому нормативну, обов'язкову та додаткову літературу. Правильно вирішив усі розрахункові / тестові завдання. Здатен виділяти суттєві ознаки вивченого за допомогою операцій синтезу, аналізу, виявляти причинно-наслідкові зв'язки, формувати висновки і узагальнення, вільно оперувати фактами та відомостями.
4	Студент достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає під час усних виступів та письмових відповідей, в основному розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому нормативну та обов'язкову літературу. Але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки. Правильно вирішив більшість розрахункових / тестових завдань. Студент здатен виділяти суттєві ознаки вивченого за допомогою операцій синтезу, аналізу, виявляти причинно-наслідкові зв'язки, у яких можуть бути окремі несуттєві помилки, формувати висновки і узагальнення, вільно оперувати фактами та відомостями.
3	Студент в цілому володіє навчальним матеріалом, викладає його основний зміст під час усних виступів та письмових розрахунків, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому окремі суттєві неточності та помилки. Правильно вирішив половину розрахункових / тестових завдань. Має ускладнення під час виділення суттєвих ознак вивченого; під час виявлення причинно-наслідкових зв'язків і формулювання висновків.

2	Студент не в повному обсязі володіє навчальним матеріалом. Фрагментарно, поверхово (без аргументації та обґрунтування) викладає його під час усних виступів та письмових розрахунків, недостатньо розкриває зміст теоретичних питань та практичних завдань, допускаючи при цьому суттєві неточності. Правильно вирішив окремі розрахункові / тестові завдання. Безсистемно відділяє випадкові ознаки вивченого; не вмiє зробити найпростіші операції аналізу і синтезу; робити узагальнення, висновки.
---	--

Методи контролю результатів навчання	Максимальна кількість балів та вимоги до їх накопичення
Практична перевірка під час лабораторних робіт	Усього 40 балів 0-5 балів: 0 – за невиконане завдання; 1-2 – за неповністю і не своєчасно виконану роботу або роботу з серйозними помилками, невірні відповіді на додаткові питання, 3-4 - за неповністю виконану роботу або роботу з деякими помилками, неповні відповіді на додаткові питання, 4,5-5 – за повне і своєчасне виконання роботи та правильні відповіді на додаткові питання
Періодичний контроль (виконання контрольних робіт)	Усього 60 балів ПК1 - 30 балів ПК2 - 30 балів: 0 – за невиконане завдання 1-5 – за не виконане, виконане епізодично завдання 6-10 – за не виконане до кінця завдання з серйозними помилками; 11-15 – частково не повне і не своєчасне виконання контрольної роботи та не правильне виконання алгоритму 16-20 - повне і своєчасне виконання контрольної роботи та правильне виконання алгоритму, з серйозними помилками 21-25 - повне і своєчасне виконання контрольної роботи та правильне виконання алгоритму, з незначними помилками 26-30 - повне і своєчасне виконання контрольної роботи та правильне виконання алгоритму

9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

ОСНОВНА ЛІТЕРАТУРА

1. Пасічник В.В., Пасічник О.В., Утрін Д.І. Веб-технології: підручник для студ. ВНЗ. Кн.1. Львів: Магнолія-2006, 2020. 335 с.
2. Янг А., Мек Б., Кантелон М. Node.js в дії. К., 2018. 432 с.
3. Brown E. Web Development with Node and Express. Sebastopol, CA: O'Reilly Media, 2019. 329 p.
4. Mardan A. Practical Node.js. Apress, 2018. 519 p.

ДОПОМІЖНА ЛІТЕРАТУРА

1. Пауерс Ш. Вивчаємо Node. Переходимо на сторону серверу. К., 2017. 304 с.

2. Teixeira P. Professional Node.js®: Building JavaScript-Based Scalable Software. Indianapolis, IN: John Wiley & Sons, 2013. 412 p.
3. Wilson J.R. Node.js the Right Way. Practical, Server-Side JavaScript That Scales. The Pragmatic Programmers, 2013. 143 p.

ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. Документація Node.js. <https://nodejs.org/uk/docs/>
2. Губачов О.П. Теорія програмування. http://dspace.pnpu.edu.ua/bitstream/123456789/3958/1/TeorProgr_GUB.pdf
3. Node.js: керівництво для фронтенд розробника. <https://codeguida.com/post/1447>
4. Підручник з прикладів Node.js MongoDB. <https://uk.csstricks.net/8225021-node-js-mongodb-tutorial-with-examples>
5. Посібник для початківців Node.js (JavaScript на сервері). <https://ua.phhsnews.com/articles/coding/beginners-guide-to-node-js-server-side-javascript.html>
6. Теорія програмування. https://www.wiki-data.uk-ua.nina.az/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F.html
7. Books about node.js. <https://github.com/giobao/books-1/tree/master/nodejs>
8. Express web framework (Node.js/JavaScript). Express web framework (Node.js/JavaScript). https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs
9. Node.js. <https://riptutorial.com/Download/node-js-ru.pdf>