



РОЗРОБКА КОМПОНЕНТНИХ ВЕБЗАСТОСУНКІВ НА ANGULAR

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 курс, весняний семестр
Обсяг дисципліни	5 кред./150 год.: 36 год. лекцій, 18 год. лабораторних занять, 96 год. СРС
Семестровий контроль/ контрольні заходи	МКР, залік
Розклад занять	https://schedule.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович. Лабораторні роботи: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

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

Дисципліна присвячена принципам компонентного програмування та практичній розробці клієнтських вебзастосунків із використанням Angular. Предметом вивчення є компонентна модель Angular, механізми зв'язування даних і взаємодії компонентів, сервіси та впровадження залежностей, архітектурна організація застосунків, маршрутизація, взаємодія з HTTP API, реактивне програмування з RxJS, форми, тестування та оптимізація продуктивності.

Метою дисципліни є формування здатності проєктувати, реалізовувати, тестувати та вдосконалювати компонентні вебзастосунки, обґрунтовано застосовуючи засоби Angular і сучасні практики організації клієнтського програмного забезпечення.

У результаті вивчення дисципліни у студентів мають сформуватися такі **компетентності**:

загальні:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- здатність вчитися та оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- здатність працювати в команді та приймати обґрунтовані рішення.

фахові:

- здатність проектувати та розробляти програмне забезпечення із застосуванням об'єктно-орієнтованого та компонентного підходів;
- здатність використовувати сучасні мови, фреймворки та інструментальні засоби для створення клієнтських вебзастосунків;
- здатність структурувати програмні системи, розподіляти відповідальність між компонентами та сервісами;
- здатність інтегрувати клієнтські застосунки із серверними API та організовувати асинхронну обробку даних;
- здатність застосовувати засоби тестування та методи підвищення якості й продуктивності програмного забезпечення.

Очікувані **результати навчання**:

- проектувати компонентну структуру Angular-застосунку та обґрунтовувати архітектурні рішення;
- створювати компоненти, шаблони, сервіси та організовувати їх взаємодію;
- застосовувати механізми зв'язування даних, життєвого циклу компонентів і впровадження залежностей;
- реалізовувати маршрутизацію, відкладене завантаження та контроль доступу;
- інтегрувати Angular-застосунок із REST API засобами HTTP-клієнта;
- застосовувати RxJS для організації асинхронних потоків даних та обробки помилок;
- реалізовувати та валідувати шаблонно-орієнтовані й реактивні форми;
- виконувати модульне тестування компонентів і сервісів та застосовувати засоби оптимізації продуктивності.

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

Пререквізити дисципліни. Вивчення дисципліни спирається на знання, отримані в дисциплінах з попередніх семестрів, зокрема:

- «Об'єктно-орієнтоване програмування» (3 семестр),
- «Веб-технології та веб-дизайн» (3 семестр),
- «Технології розробки програмного забезпечення» (5 семестр).

Постреквізити дисципліни. Матеріал дисципліни може бути використаний при вивченні дисциплін: «Вступ до інтелектуального аналізу даних», «Моделювання систем в енергетиці», «Технології паралельних обчислень в енергетичних комплексах», «Проектування інформаційних систем» та інших, що подаються в наступних семестрах, а також при дипломному проектуванні.

3. Зміст навчальної дисципліни

Тема 1. Односторінкові вебзастосунки, Angular та TypeScript.

Тема 2. Компонентна модель та механізми взаємодії компонентів.

Тема 3. Сервіси, впровадження залежностей та архітектурна організація Angular-застосунків.

Тема 4. Маршрутизація та організація функціональних частин застосунку.

Тема 5. Взаємодія з серверними API та реактивне програмування.

Тема 6. Форми та валідація даних.

Тема 7. Тестування та оптимізація Angular-застосунків.

4. Навчальні матеріали та ресурси

Основна література

1. Freeman A. *Pro Angular: Build Powerful and Dynamic Web Apps*. 5th ed. Berkeley, CA : Apress, 2022. 880 p.
2. Bampakos A. *Angular Projects: Build Modern Web Apps in Angular 16 with 10 Different Projects and Cutting-Edge Technologies*. 3rd ed. Birmingham : Packt Publishing, 2023. 312 p.

3. Angular Documentation [Електронний ресурс] / Google. Режим доступу: <https://angular.dev/> (дата звернення: 01.06.2026).
4. TypeScript Handbook [Електронний ресурс] / Microsoft. Режим доступу: <https://www.typescriptlang.org/docs/handbook> (дата звернення: 01.06.2026).
5. RxJS Documentation [Електронний ресурс]. Режим доступу: <https://rxjs.dev/guide> (дата звернення: 01.06.2026).

Додаткова література

1. Martin R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Boston : Pearson, 2017. 432 p.
2. Fowler M. *Patterns of Enterprise Application Architecture*. Boston : Addison-Wesley Professional, 2002. 560 p.
3. Mozilla Developer Network. MDN Web Docs [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/> (дата звернення: 01.06.2026).
4. TypeScript Documentation [Електронний ресурс] / Microsoft. Режим доступу: <https://www.typescriptlang.org/docs> (дата звернення: 01.06.2026).

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

5.1 Лекційні заняття

Тема 1. Односторінкові вебзастосунки, Angular та TypeScript

Лекція 1. Вступ до односторінкових вебзастосунків та фреймворку Angular.

Еволюція вебзастосунків та підходів до їх побудови. Порівняння багатосторінкових і односторінкових застосунків. Концепція SPA. Роль JavaScript-фреймворків. Загальна характеристика Angular, його можливості та сфери застосування.

Лекція 2. Архітектура Angular-застосунку та принципи його організації.

Загальна архітектурна модель Angular-застосунку. Angular CLI. Типова структура проекту. Роль компонентів, сервісів і механізму впровадження залежностей. Принципи структурування коду.

Лекція 3. Використання TypeScript у розробці Angular-застосунків.

Система типів TypeScript. Інтерфейси та класи. Декоратори. Узагальнені типи. Асинхронне програмування. Практичні аспекти використання TypeScript в Angular.

Тема 2. Компонентна модель та механізми взаємодії компонентів

Лекція 4. Компонентна модель Angular-застосунку.

Поняття компонента. Структура та метадані компонента. Шаблони. Стилзація та інкапсуляція. Повторне використання та ієрархія компонентів.

Лекція 5. Механізми зв'язування даних у Angular.

Концепція зв'язування даних. Інтерполяція. Прив'язка властивостей і подій. Двостороннє зв'язування. Типові сценарії використання та обмеження.

Лекція 6. Життєвий цикл компонентів Angular.

Фази життєвого циклу компонента. Основні методи життєвого циклу, порядок їх виклику та практичні сценарії використання. Керування ресурсами компонента.

Лекція 7. Організація взаємодії між компонентами.

Передача даних між батьківськими та дочірніми компонентами. Вхідні та вихідні властивості. Події компонентів. Доступ до дочірніх елементів. Підходи до організації взаємодії.

Тема 3. Сервіси, впровадження залежностей та архітектурна організація

Лекція 8. Сервіси та механізм впровадження залежностей.

Призначення сервісів. Принципи Dependency Injection. Провайдери, області видимості та життєвий цикл сервісів. Відокремлення логіки представлення від прикладної логіки.

Лекція 9. Архітектурні підходи до побудови Angular-застосунків.

Розділення відповідальностей. Презентаційні та контейнерні компоненти. Організація застосунку за функціональними частинами. Структурування великих застосунків. Типові архітектурні помилки.

Тема 4. Маршрутизація та організація функціональних частин застосунку

Лекція 10. Маршрутизація в Angular-застосунках.

Поняття маршрутизації у SPA. Конфігурація маршрутів. Програмна і декларативна навігація. Параметри та вкладені маршрути.

Лекція 11. Відкладене завантаження та контроль доступу до маршрутів.

Концепція lazy loading. Декомпозиція застосунку за функціональними частинами. Механізми захисту маршрутів і контроль доступу. Вплив відкладеного завантаження на продуктивність.

Тема 5. Взаємодія з серверними API та реактивне програмування

Лекція 12. Взаємодія Angular-застосунку з серверними API.

Роль HTTP у клієнтських застосунках. Засоби HTTP-клієнта Angular. CRUD-операції. Заголовки та параметри запитів. Обробка відповідей. Перехоплювачі HTTP-запитів.

Лекція 13. Основи реактивного програмування у Angular-застосунках.

Поняття реактивного програмування. Потoki даних та асинхронна обробка подій. Observable і підписки. Керування життєвим циклом підписок. Порівняння Observable та Promise.

Лекція 14. Оператори перетворення потоків даних та обробка помилок.

Класифікація операторів RxJS. Перетворення, фільтрація та комбінування потоків. Організація асинхронних ланцюжків. Обробка помилок і повторне виконання операцій.

Тема 6. Форми та валідація даних

Лекція 15. Реалізація форм на основі шаблонного підходу.

Форми у вебзастосунках. Шаблонно-орієнтований підхід. Директиви форм. Зв'язування даних. Валідація користувацького вводу. Обмеження підходу.

Лекція 16. Реалізація форм на основі реактивного підходу.

Концепція реактивних форм. FormControl, FormGroup і FormArray. Керування станом. Синхронна та асинхронна валідація. Динамічні форми. Порівняння підходів.

Тема 7. Тестування та оптимізація Angular-застосунків

Лекція 17. Методи та засоби тестування клієнтських застосунків.

Роль тестування. Види тестування клієнтських застосунків. Тестове середовище Angular. Тестування компонентів і сервісів. Ізоляція залежностей і тестові двійники.

Лекція 18. Оптимізація продуктивності та узагальнення підходів до розробки Angular-застосунків.

Продуктивність клієнтських застосунків. Механізм виявлення змін. Оптимізація рендерингу. Стратегії завантаження ресурсів і розбиття коду. Типові помилки та узагальнення архітектурних практик.

5.2 Лабораторні роботи

Лабораторна робота 1. Створення та базова організація Angular-проєкту. (2 год)

Створення застосунку засобами Angular CLI, аналіз структури проєкту, налаштування середовища розробки та формування базової структури застосунку.

Лабораторна робота 2. Розробка компонентів, шаблонів та механізмів зв'язування даних.

Побудова ієрархії компонентів, формування шаблонів і стилів, реалізація інтерполяції, прив'язки властивостей і подій, двостороннього зв'язування даних та використання механізмів життєвого циклу компонентів.

Лабораторна робота 3. Взаємодія компонентів, сервіси та впровадження залежностей.
(2 год)

Організація обміну даними та подіями між компонентами, створення сервісів, використання механізму Dependency Injection та розподіл відповідальності між компонентами і сервісами.

Лабораторна робота 4. Архітектурна організація та маршрутизація Angular-застосунку.

Структурування застосунку за функціональними частинами, налаштування маршрутів і параметрів навігації, реалізація відкладеного завантаження та контролю доступу до маршрутів.

Лабораторна робота 5. Взаємодія з серверним API та реактивна обробка даних.

Реалізація CRUD-операцій засобами HTTP-клієнта Angular, організація сервісного шару, створення та перетворення потоків даних засобами RxJS, керування підписками та обробка помилок.

Лабораторна робота 6. Реалізація та валідація форм.

Побудова шаблонно-орієнтованих і реактивних форм, програмне керування станом форми, реалізація синхронної та асинхронної валідації користувацького вводу.

Лабораторна робота 7. Тестування компонентного Angular-застосунку.

Модульне тестування компонентів і сервісів, ізоляція залежностей, аналіз продуктивності та застосування базових прийомів оптимізації Angular-застосунку.

6. Самостійна робота студента

Тема 1. Опрацювання сучасних підходів до побудови SPA та можливостей інструментального середовища Angular.

Тема 2. Розроблення повторно використовуваних компонентів та дослідження механізмів їх взаємодії.

Тема 3. Порівняння підходів до архітектурної організації клієнтських застосунків.

Тема 4. Дослідження стратегій маршрутизації, відкладеного завантаження та контролю доступу.

Тема 5. Опрацювання засобів RxJS для складних сценаріїв асинхронної взаємодії.

Тема 6. Порівняння шаблонно-орієнтованих і реактивних форм та розроблення власних валідаторів.

Тема 7. Підготовка індивідуального компонентного Angular-застосунку та його тестування.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни (на кожну роботу відводиться два тижні для здачі),
- пишуть модульну контрольну роботу.

Виконання та захист лабораторних робіт є обов'язковою складовою опанування дисципліни. На виконання кожної лабораторної роботи відводиться два тижні. У разі пропуску заняття з поважної причини студенту надається можливість виконати та захистити відповідну роботу в узгоджений із викладачем термін.

Лабораторні роботи виконуються в межах наскрізного семестрового проєкту. Кожна наступна лабораторна робота розвиває результати попередніх робіт. Студент повинен своєчасно виконувати завдання, демонструвати отримані результати та захищати прийняті проєктні й програмні рішення.

Несвоєчасне виконання лабораторних робіт без поважної причини може враховуватися під час оцінювання відповідно до рейтингової системи. Під час виконання робіт студент повинен

дотримуватися принципів академічної доброчесності. Використання зовнішніх програмних засобів, бібліотек та генеративних систем допускається відповідно до завдання і вимог викладача із розумінням та здатністю пояснити створене рішення.

Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Передбачена можливість зарахування результатів навчання за окремими темами дисципліни за результатами успішного проходження онлайн-курсів. Визнання таких результатів здійснюється відповідно до «Положення про визнання в КПІ ім. Ігоря Сікорського результатів навчання, набутих у неформальній/інформальній освіті» (<https://osvita.kpi.ua/node/179>).

Оскільки лабораторні роботи виконуються в межах наскрізного семестрового проєкту, зарахування результатів онлайн-навчання не звільняє студента від виконання тих складових проєкту, які необхідні для забезпечення його цілісності та виконання наступних лабораторних робіт.

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

Система рейтингових (вагових) балів та критерії оцінювання

1) Лабораторні роботи

Максимальна кількість балів за лабораторну роботу визначається її обсягом і складністю та становить 10 або 15 балів відповідно до таблиці вагових коефіцієнтів.

Для оцінки роботи використовується наступні критерії оцінювання:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру лабораторної роботи;
- виконаний із запізненням – знімається 10–30% від максимальної кількості балів в залежності від терміну запізнення;
- виконаний не самостійно, із запізненням – знімається 50% від максимальної кількості балів;
- невиконаний протягом відведеного часу – 0 балів.

Максимальна кількість балів за всі виконані лабораторні роботи становить 90 балів.

Розподіл рейтингових балів за лабораторні роботи наведено в таблиці.

№ з/п	Назва лабораторної роботи	Ваговий коефіцієнт
1	Створення та базова організація Angular-проєкту	10
2	Розробка компонентів, шаблонів та механізмів зв'язування даних	15
3	Взаємодія компонентів, сервіси та впровадження залежностей	15
4	Архітектурна організація та маршрутизація Angular-застосунку	15
5	Взаємодія з серверним API та реактивна обробка даних	15
6	Реалізація та валідація форм	10
7	Тестування компонентного Angular-застосунку	10

Передбачено систему штрафних та заохочувальних балів:

активність на лабораторних роботах	+ 1 бали
виконання лабораторної роботи: • з використанням власного архітектурного рішення з дотриманням сучасних стандартів програмування; • застосовано обґрунтоване архітектурне або компонентне рішення, що виходить за межі базових вимог лабораторної роботи;	+ 0,5 бали

- реалізовано додаткову функціональність Angular-застосунку, що виходить за межі базового завдання та відповідає сучасним практикам розробки; - покриття модульними тестами виконано на високому рівні.	
несвоєчасна здача лабораторної роботи без поважної причини	-0,5 балів- -1 бал
обидві наскрізні атестації позитивні	+2 бал

Загальна сума рейтингових балів з урахуванням заохочувальних балів не може перевищувати 100 балів.

2) Календарний контроль

Проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Для отримання «зараховано» з першої проміжної атестації студент повинен захистити не менше 2 лабораторних робіт.

Для отримання «зараховано» з другої проміжної атестації студент повинен захистити не менше 4 лабораторних робіт та набрати не менше 30 балів.

3) Модульна контрольна робота

Модульна контрольна робота (МКР) проводиться на лекційному занятті 18-го тижня з метою контролю рівня засвоєння матеріалу дисципліни.

Максимальна кількість балів за МКР дорівнює 10 балів.

Якість виконання роботи оцінюється за наступними критеріями:

- усі відповіді вірні та повні – 10 балів,
- у відповідях допущені несуттєві неточності – 8 балів,
- половина відповідей вірна – 5 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 2 бали,
- менше половини відповідей вірна – 0 балів.

Можливе також проведення модульної контрольної роботи у формі тестування, в такому випадку студент отримує бал, який відповідає частці правильних відповідей тесту з розрахунку, що 10 балів становлять 100%.

4) Семестровий контроль (залік)

Залік виставляється за результатами роботи студента протягом семестру відповідно до набраного рейтингу. Умовою отримання заліку є виконання та захист усіх лабораторних робіт, виконання модульної контрольної роботи та отримання не менше 60 рейтингових балів.

5) Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{практ}} + r_{\text{мод}} = 90 + 10 = 100 \text{ балів.}$$

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно

R < 40 є незараховані лабораторні роботи або не виконані інші умови допуску до семестрового контролю	Не допущено
--	--------------------

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович

Ухвалено кафедрою ЦТЕ (протокол № 22 від 17.06.26)

Погоджено науково-методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 26.06.2026 р.)