

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

МЕТОДИЧНІ ВКАЗІВКИ
до виконання самостійних робіт
з дисципліни «Програмування та підтримка веб-застосунків»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
124 Системний аналіз
(частина 2)

Одеса – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

МЕТОДИЧНІ ВКАЗІВКИ
до виконання самостійних робіт
з дисципліни «Програмування та підтримка веб-застосунків»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
124 Системний аналіз
(частина 2)

Затверджено
на засіданні кафедри прикладної
математики та інформаційних
технологій
Протокол № _ від _____

Одеса – 2025

Методичні вказівки до виконання самостійних робіт з дисципліни «Програмування та підтримка веб-застосунків» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 Системний аналіз (частина 2) / Укл.: К. О. Трифонова. – Одеса, 2025. – 13 с.

Укладач: Трифонова К. О., ст. викл.

ЗМІСТ

Вступ	5
Самостійна робота №9. Представлення	6
Самостійна робота №10 Компоненти представлень	7
Самостійна робота №11. Допоміжні функції дескрипторів	8
Самостійна робота №12. Допоміжні функції дескрипторів для форм	9
Самостійна робота №13. Спеціалізовані допоміжні функції дескрипторів	10
Самостійна робота №14. Прив'язка моделей	11
Самостійна робота №15. Перевірка достовірності моделей	12

Вступ

Навчальна дисципліна «Програмування та підтримка веб-застосувань» є необхідною дисципліною для підвищення рівня теоретичних і прикладних знань, що формують фахівця в галузі інформаційних технологій, які сприяють утворенню у здобувачів поглиблених вмінь та навичок веб-розробки програмного забезпечення із застосуванням сучасних парадигм програмування, тобто підготовці спеціалістів для створення, впровадження та підтримки професійно-орієнтованих комп'ютерних технологій у професійній діяльності.

Дисципліна «Програмування та підтримка веб-застосувань» відповідає освітньо-професійній програмі, навчальному плану підготовки фахівців першого (бакалаврського) рівня вищої освіти спеціальності 124 Системний аналіз і є складовою циклу дисциплін професійної підготовки вибіркової частини навчального плану.

Дисципліну викладають впродовж третього семестру першого (бакалаврського) рівня. У процесі навчання передбачено лекції, лабораторні заняття.

Згідно навчального плану передбачено підсумковий контроль у вигляді заліку.

Робочу програму навчальної дисципліни укладено згідно з вимогами кредитно-модульної системи організації навчального процесу. Програма визначає обсяги компетентностей, які повинен опанувати здобувач відповідно до освітньо-професійної програми, алгоритму вивчення навчального матеріалу дисципліни «Програмування та підтримка веб-застосувань», необхідне методичне забезпечення, складові та технологію оцінювання навчальних досягнень.

Предмет навчальної дисципліни «Програмування та підтримка веб-застосувань» – сучасні технології веб-розробки програмного забезпечення.

Мета вивчення навчальної дисципліни «Програмування та підтримка веб-застосувань» – забезпечити формування поглибленої системи теоретичних і практичних знань у галузі веб-розробки програмного забезпечення для реалізації здатності використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

Завдання вивчення дисципліни:

- формувати теоретичні основи та практичні навички застосування методів вилучення, аналізу, обробки та синтезу інформації для формування технічного завдання при розв'язанні задач в галузі інформаційних технологій;
- формувати теоретичні основи та практичні навички сучасних парадигм веб-програмування та способів їх вибору з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інформаційних технологій;
- поглибити практичні навички кодування веб-програмного забезпечення для розв'язання задач в галузі інформаційних технологій;
- поглибити практичні навички налагодження веб-програмного забезпечення для розв'язання задач в галузі інформаційних технологій.

Стратегічні цілі дисципліни – націлити майбутніх фахівців на творче застосування отриманих знань у подальшій професійній підготовці та їх наступній практичній діяльності.

САМОСТІЙНА РОБОТА №9. ПРЕДСТАВЛЕННЯ

1. Дайте визначення поняттю представлення для шаблону програмного забезпечення MVC.
2. Дайте визначення поняттю механізм візуалізації для застосування ASP.NET Core MVC.
3. Поясніть необхідність розробки спеціального механізму візуалізації для застосування ASP.NET Core MVC.
4. Поясніть призначення класу, який реалізує інтерфейс `IView`, для розробки спеціального механізму візуалізації для застосування ASP.NET Core MVC.
5. Поясніть призначення класу, який реалізує інтерфейс `ViewEngine`, для розробки спеціального механізму візуалізації для застосування ASP.NET Core MVC.
6. Поясніть необхідність реєстрації спеціального механізму візуалізації для застосування ASP.NET Core MVC.
7. Поясніть механізм функціонування спеціального механізму візуалізації для застосування ASP.NET Core MVC.
8. Назвіть вбудований механізм візуалізації для застосування ASP.NET Core MVC.
9. Поясніть механізм функціонування вбудованого механізму візуалізації для застосування ASP.NET Core MVC.
10. Назвіть базовий клас для класу представлення, який формується вбудованим механізмом візуалізації для застосування ASP.NET Core MVC.
11. Поясніть призначення статичного вмісту для представлення для застосування ASP.NET Core MVC.
12. Поясніть призначення динамічного вмісту для представлення для застосування ASP.NET Core MVC.
13. Поясніть призначення компоновки для представлення для застосування ASP.NET Core MVC.
14. Поясніть призначення розділу представлення для компоновки для застосування ASP.NET Core MVC.
15. Поясніть необхідність виконання перевірити існування розділу представлення у компоновці для застосування ASP.NET Core MVC.
16. Поясніть необхідність використання необов'язкового розділу представлення у компоновці для застосування ASP.NET Core MVC.
17. Поясніть призначення часткових представлень для застосування ASP.NET Core MVC.
18. Поясніть необхідність використання у представленні включення даних у форматі JSON у відповідь, що надсилається браузеру для застосування ASP.NET Core MVC.
19. Поясніть стандартний механізм функціонування для вбудованого механізму візуалізації пошуку розташування представлення для застосування ASP.NET Core MVC.
20. Поясніть необхідність розширення розташування представлень для вбудованого механізму візуалізації для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №10 КОМПОНЕНТИ ПРЕДСТАВЛЕНЬ

1. Дайте визначення поняттю компонента представлення для застосування ASP.NET Core MVC.
2. Поясніть необхідність використання компоненти представлення для застосування ASP.NET Core MVC.
3. Перерахуйте способи створення компоненти представлення для застосування ASP.NET Core MVC.
4. Дайте визначення поняттю компонента представлення POCO для застосування ASP.NET Core MVC.
5. Поясніть особливості реалізації компоненти представлення POCO для застосування ASP.NET Core MVC.
6. Дайте визначення поняттю компонента представлення, яка є похідним класом від класу ViewComponent для застосування ASP.NET Core MVC.
7. Поясніть переваги використання компоненти представлення, яка є похідним класом від класу ViewComponent для застосування ASP.NET Core MVC.
8. Назвіть клас, який реалізує інтерфейс IViewComponentResult, екземпляр якого повертається компонентою представлення для повідомлення механізму Razor про необхідність візуалізації часткового представлення для застосування ASP.NET Core MVC.
9. Поясніть механізм функціонування для механізму Razor пошуку розташування часткового представлення, що запитується компонентою представлення для застосування ASP.NET Core MVC.
10. Назвіть клас, який реалізує інтерфейс IViewComponentResult, екземпляр якого повертається компонентою представлення для включення фрагментів HTML-розмітки до батьківського представлення, не використовуючи представлення для застосування ASP.NET Core MVC.
11. Поясніть необхідність повернення фрагмента надійної HTML-розмітки компонентою представлення для застосування ASP.NET Core MVC.
12. Перерахуйте властивості класу ViewComponentContext, які дозволяють отримати дані контексту для компоненти представлення для застосування ASP.NET Core MVC.
13. Перерахуйте властивості класу ViewComponent, які дозволяють отримати дані контексту для компоненти представлення для застосування ASP.NET Core MVC.
14. Поясніть необхідність використання асинхронної компоненти представлення для застосування ASP.NET Core MVC.
15. Поясніть механізм функціонування асинхронної компоненти представлення для застосування ASP.NET Core MVC.
16. Поясніть необхідність використання гібридного класу контролера та компоненти представлення для застосування ASP.NET Core MVC.
17. Поясніть недолік реалізації гібридного класу контролера та компоненти представлення для застосування ASP.NET Core MVC.
18. Поясніть механізм функціонування асинхронного гібридного класу контролера та компоненти представлення для застосування ASP.NET Core MVC.
19. Перерахуйте типи даних класів, які може повертати компонента представлення, як складова гібридного класу контролера та компоненти представлення для застосування ASP.NET Core MVC.
20. Перерахуйте складові проекту, для яких може бути застосований гібридний клас контролера та компоненти представлення для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №11. ДОПОМІЖНІ ФУНКЦІЇ ДЕСКРИПТОРІВ

1. Поясніть призначення допоміжної функції дескриптора для застосування ASP.NET Core MVC.
2. Назвіть базовий клас і метод, який повинен бути перевизначений, для класу допоміжної функції дескриптора для застосування ASP.NET Core MVC.
3. Поясніть призначення класу TagHelperContext і перерахуйте кілька його членів для застосування ASP.NET Core MVC.
4. Поясніть призначення класу TagHelperOutput і перерахуйте кілька його членів для застосування ASP.NET Core MVC.
5. Поясніть способи реєстрації допоміжної функції дескриптора для застосування ASP.NET Core MVC.
6. Поясніть спосіб використання допоміжної функції дескриптора для застосування ASP.NET Core MVC.
7. Поясніть необхідність застосування звуження області дії для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
8. Поясніть необхідність застосування розширення області дії для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
9. Поясніть призначення класу допоміжної функції дескриптора, який створює дескриптор для скорочення для застосування ASP.NET Core MVC.
10. Поясніть призначення класу TagHelperContent і перерахуйте кілька його членів для застосування ASP.NET Core MVC.
11. Поясніть механізм додавання вмісту навколо цільового дескриптора, в результаті створення класу допоміжної функції дескриптора для застосування ASP.NET Core MVC.
12. Поясніть механізм додавання вмісту в цільовий дескриптор, в результаті створення класу допоміжної функції дескриптора для застосування ASP.NET Core MVC.
13. Поясніть необхідність використання впровадження залежностей для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
14. Поясніть механізм використання впровадження залежностей для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
15. Поясніть необхідність використання моделі представлення для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
16. Поясніть призначення класу ModelExpression і перерахуйте кілька його членів для застосування ASP.NET Core MVC.
17. Поясніть необхідність реалізації узгодження допоміжних функцій дескриптора для застосування ASP.NET Core MVC.
18. Поясніть необхідність використання властивості TagHelperContext.Items для допоміжної функції дескриптора для застосування ASP.NET Core MVC.
19. Поясніть необхідність запобігання включення дескриптора за допомогою допоміжної функції дескриптора для застосування ASP.NET Core MVC.
20. Поясніть механізм функціонування запобігання включення дескриптора за допомогою допоміжної функції дескриптора для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №12. ДОПОМІЖНІ ФУНКЦІЇ ДЕСКРИПТОРІВ ДЛЯ ФОРМ

1. Поясніть призначення допоміжних функцій дескрипторів форм для застосування ASP.NET Core MVC.
2. Назвіть клас, який реалізує допоміжні функції дескриптора form для застосування ASP.NET Core MVC.
3. Поясніть переваги використання допоміжних функцій дескриптора form asp-action та asp-controller для застосування ASP.NET Core MVC.
4. Назвіть механізм, за допомогою якого організовано ідентифікацію запитів, що відносяться до специфічного сеансу, з яким зазвичай асоційований користувач для застосування ASP.NET Core MVC.
5. Поясніть механізм функціонування атаки підробки міжсайтових запитів для застосування ASP.NET Core MVC.
6. Поясніть необхідність використання атрибуту asp-for для дескриптора input для застосування ASP.NET Core MVC.
7. Перерахуйте атрибути для HTML-розмітки, які генеруються в результаті використання представлення, що містить форму з дескриптором input та атрибутом asp-for для застосування ASP.NET Core MVC.
8. Поясніть відповідність між типами даних C# та значеннями атрибуту type дескриптора input для застосування ASP.NET Core MVC.
9. Поясніть необхідність використання атрибуту UIHint для моделі представлення для застосування ASP.NET Core MVC.
10. Поясніть необхідність використання атрибуту asp-format для дескриптора input для застосування ASP.NET Core MVC.
11. Поясніть перевагу використання атрибуту DisplayFormat для властивості моделі представлення для застосування ASP.NET Core MVC.
12. Поясніть необхідність використання атрибуту asp-for для дескриптора label для застосування ASP.NET Core MVC.
13. Перерахуйте атрибути для HTML-розмітки, які генеруються в результаті використання представлення, що містить форму з дескриптором label та атрибутом asp-for для застосування ASP.NET Core MVC.
14. Назвіть клас, який реалізує допоміжні функції дескриптора select для застосування ASP.NET Core MVC.
15. Перерахуйте атрибути для розмітки HTML, які генеруються в результаті використання представлення, що містить форму з дескриптором select і атрибутом asp-for для застосування ASP.NET Core MVC.
16. Перерахуйте можливі джерела даних, які використовуються при генерації дескрипторів option для дескриптора select для застосування ASP.NET Core MVC.
17. Поясніть, можливі проблеми, що виникають при використанні ViewBag в якості джерела даних при генерації дескрипторів option для дескриптора select для застосування ASP.NET Core MVC.
18. Назвіть клас, який реалізує допоміжні функції дескриптора textarea для застосування ASP.NET Core MVC.
19. Перерахуйте атрибути для розмітки HTML, які генеруються в результаті використання представлення, що містить форму з дескриптором textarea і атрибутом asp-for для застосування ASP.NET Core MVC.
20. Поясніть необхідність використання допоміжних функцій дескрипторів для перевірки достовірності форм для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №13. СПЕЦІАЛІЗОВАНІ ДОПОМІЖНІ ФУНКЦІЇ ДЕСКРИПТОРІВ

1. Поясніть призначення допоміжної функції дескриптора для середовища розміщення для застосування ASP.NET Core MVC.
2. Назвіть клас, який реалізує допоміжні функції дескриптора для середовища розміщення для застосування ASP.NET Core MVC.
3. Назвіть клас, який реалізує допоміжні функції дескриптора для управління включенням файлів JavaScript у представлення для застосування ASP.NET Core MVC.
4. Поясніть необхідність використання в компоновці атрибуту допоміжної функції дескриптора включення файлів JavaScript у представлення, з використанням шаблону універсалізації імен для застосування ASP.NET Core MVC.
5. Поясніть необхідність використання в компоновці атрибуту допоміжної функції дескриптора виключення файлів JavaScript у представленні, з використанням шаблону універсалізації імен для застосування ASP.NET Core MVC.
6. Поясніть перевагу використання мережі доставки вмісту для застосування ASP.NET Core MVC.
7. Назвіть клас, який реалізує допоміжні функції дескриптора для управління включенням файлів CSS у представлення для застосування ASP.NET Core MVC.
8. Поясніть необхідність використання в компоновці атрибуту допоміжної функції дескриптора включення файлів CSS у представлення, з використанням шаблону універсалізації імен для застосування ASP.NET Core MVC.
9. Поясніть необхідність використання в компоновці атрибуту допоміжної функції дескриптора виключення файлів CSS у представленні, з використанням шаблону універсалізації імен для застосування ASP.NET Core MVC.
10. Поясніть недоліки використання мережі доставки вмісту для застосування ASP.NET Core MVC.
11. Назвіть клас, який реалізує допоміжні функції дескриптора для навігації для застосування ASP.NET Core MVC.
12. Назвіть клас, який реалізує допоміжні функції дескриптора для зображення для застосування ASP.NET Core MVC.
13. Поясніть необхідність використання кешування фрагментів вмісту для застосування ASP.NET Core MVC.
14. Назвіть клас, який реалізує допоміжні функції дескриптора для управління кешуванням фрагментів вмісту для застосування ASP.NET Core MVC.
15. Поясніть необхідність використання анулювання кешування фрагментів вмісту для застосування ASP.NET Core MVC.
16. Поясніть механізм функціонування анулювання кешування фрагментів вмісту для застосування ASP.NET Core MVC.
17. Поясніть необхідність встановлення періоду закінчення часу для кешу вмісту представлення для застосування ASP.NET Core MVC.
18. Поясніть необхідність встановлення фіксованого часу для кешу вмісту представлення для застосування ASP.NET Core MVC.
19. Поясніть необхідність встановлення періоду закінчення часу з моменту останнього використання для кешу вмісту представлення для застосування ASP.NET Core MVC.
20. Поясніть необхідність встановлення періоду закінчення часу для кешу вмісту представлення для кожного запиту для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №14. Прив'язка моделей

1. Дайте визначення поняттю прив'язка моделей для застосування ASP.NET Core MVC.
2. Дайте визначення поняттю зв'язувачі моделей для застосування ASP.NET Core MVC.
3. Поясніть спосіб призначення значення моделі прив'язкою моделей для методу дії, у випадку, якщо користувач не заповнив дані форми, не передав сегменти строчки запиту, у шаблоні маршрутизації не вказані значення за замовчуванням для застосування ASP.NET Core MVC.
4. Поясніть наслідки відсутності можливості отримання значення моделі прив'язкою моделей для методу дії для застосування ASP.NET Core MVC.
5. Перерахуйте прості типи даних, які можуть бути використані для моделей прив'язкою моделей для застосування ASP.NET Core MVC.
6. Поясніть механізм функціонування прив'язки моделей для моделей простих типів даних для застосування ASP.NET Core MVC.
7. Перерахуйте складні типи даних, які можуть бути використані для моделей прив'язкою моделей для застосування ASP.NET Core MVC.
8. Поясніть механізм функціонування прив'язки моделей для складних типів даних для застосування ASP.NET Core MVC.
9. Поясніть механізм функціонування прив'язки моделей для типів даних масивів для застосування ASP.NET Core MVC.
10. Поясніть механізм функціонування прив'язки моделей для типів даних колекцій для застосування ASP.NET Core MVC.
11. Поясніть механізм функціонування прив'язки моделей для моделей типів даних колекцій складних типів для застосування ASP.NET Core MVC.
12. Поясніть механізм візуалізації представлення, в якості моделі якого визначається колекція складних типів даних для застосування ASP.NET Core MVC.
13. Поясніть необхідність використання атрибутів для джерел даних прив'язки моделей для застосування ASP.NET Core MVC.
14. Поясніть необхідність зміни стандартного джерела даних прив'язки моделей для застосування ASP.NET Core MVC.
15. Поясніть спосіб зміни стандартного джерела даних прив'язки моделей для застосування ASP.NET Core MVC.
16. Поясніть спосіб зміни порядку стандартного джерела даних прив'язки моделей для застосування ASP.NET Core MVC.
17. Поясніть спосіб звернення до конкретного імені заголовка HTTP-запиту, який використовується в якості джерела даних прив'язки моделей для застосування ASP.NET Core MVC.
18. Поясніть переваги використання атрибутів для властивостей моделі, які вказують в якості джерела даних імена заголовка HTTP-запиту для прив'язки моделей для застосування ASP.NET Core MVC.
19. Поясніть необхідність використання тіла HTTP-запиту в якості джерела даних прив'язки моделей для застосування ASP.NET Core MVC.
20. Поясніть спосіб використання атрибутуFromBody для кількох параметрів методу дії, в якості джерела даних прив'язки моделей для застосування ASP.NET Core MVC.

САМОСТІЙНА РОБОТА №15. ПЕРЕВІРКА ДОСТОВІРНОСТІ МОДЕЛЕЙ

1. Дайте визначення поняттю перевірка достовірності моделі для застосування ASP.NET Core MVC.
2. Поясніть необхідність виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
3. Поясніть призначення класу ModelStateDictionary та перерахуйте кілька його членів для застосування ASP.NET Core MVC.
4. Поясніть призначення перерахування ModelState і перерахуйте кілька його значень для застосування ASP.NET Core MVC.
5. Поясніть функціонування механізму призначення стилізованих змін полів введення представлення, за допомогою таблиць стилів CSS, в результаті виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
6. Поясніть функціонування механізму призначення стилізованих змін полів введення представлення, за допомогою бібліотеки Bootstrap, в результаті виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
7. Поясніть функціонування механізму відображення звіту інформаційних повідомлень для представлення, в результаті виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
8. Поясніть призначення класу DefaultModelBindingMessageProvider і перерахуйте кілька його членів для застосування ASP.NET Core MVC.
9. Поясніть функціонування механізму відображення інформаційного повідомлення поруч із полем введення для представлення, в результаті виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
10. Поясніть необхідність використання звіту інформаційних повідомлень для представлення, в результаті виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
11. Поясніть призначення перерахування ValidationSummary та перерахуйте кілька його значень для застосування ASP.NET Core MVC.
12. Поясніть переваги використання атрибутів у моделі для виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
13. Перерахуйте кілька вбудованих атрибутів моделі для виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
14. Поясніть необхідність розробки класу, який реалізує спеціальний атрибут для виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
15. Назвіть інтерфейс, який повинен бути реалізований класом, який реалізує спеціальний атрибут для виконання перевірки достовірності моделі для застосування ASP.NET Core MVC.
16. Поясніть переваги виконання перевірки достовірності моделі на стороні сервера для застосування ASP.NET Core MVC.
17. Поясніть переваги виконання перевірки достовірності моделі на стороні клієнта для застосування ASP.NET Core MVC.
18. Поясніть функціонування механізму віддаленої перевірки достовірності моделі на стороні клієнта для застосування ASP.NET Core MVC.
19. Поясніть переваги виконання віддаленої перевірки достовірності моделі на стороні клієнта для застосування ASP.NET Core MVC.
20. Поясніть основну відмінність виконання віддаленої перевірки достовірності моделі на стороні клієнта від перевірки достовірності моделі на стороні сервера для застосування ASP.NET Core MVC.

Методичні вказівки до виконання самостійних робіт з дисципліни «Програмування та підтримка веб-застосунків» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 Системний аналіз (частина 2) / Укл.: К. О. Трифонова. – Одеса, 2025. – 13 с.

Укладач: Трифонова К. О., ст. викл.

Підписано до друку _____. Формат 60x84/16. Папір газетний. Друк офсетний. 0,87
ум. друк. арк. 0,94 обл. - вид. арк.
Тираж 100 пр. Зам. №

Національний університет «Одеська політехніка»
65044, Одеса, пр. Шевченка, 1