

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ**

Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра інженерії програмного забезпечення та кібербезпеки

ЗАТВЕРДЖЕНО

вченою радою ДТЕУ

(пост. п. від «24» 06 2024 р.)

Ректор

Анатолій МАЗАРАКІ



**ТЕХНОЛОГІЇ РОЗРОБКИ ТА ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ/
TECHNOLOGIES OF DEVELOPMENT AND
TESTING OF SOFTWARE**

**ПРОГРАМА /
COURSE SUMMARY**

Київ 2024

**Розповсюдження і тиражування без офіційного дозволу ДТЕУ
заборонено**

Автори: Т.О. ЖИРОВА, кандидат педагогічних наук, доцент
кафедри інженерії програмного забезпечення та
кібербезпеки
Н.О. КОТЕНКО, кандидат педагогічних наук, доцент
кафедри інженерії програмного забезпечення та
кібербезпеки
Я.Р. ЯРЕМИЧ, асистент

Програму розглянуто і затверджено на засіданні кафедри інженерії
програмного забезпечення та кібербезпеки «З» червня 2024р., протокол
№ 36.

Рецензенти: Т.В. Савченко, к.т.н., доцент кафедри інженерії
програмного забезпечення та кібербезпеки;
О.О. РУДЕНКО, SFCC Front-End Team Lead, Raccoon
LLC

**ТЕХНОЛОГІЇ РОЗРОБКИ ТА ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ/
TECHNOLOGIES OF DEVELOPMENT AND
TESTING OF SOFTWARE**

**ПРОГРАМА /
COURSE SUMMARY**

ВСТУП

Програма дисципліни «Технології розробки та тестування програмного забезпечення» галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Комп'ютерні науки».

Програму підготовлено відповідно до Стандарту вищої освіти України за даною спеціальністю та відповідної освітньо-професійної програми підготовки магістрів.

Програма складається з таких розділів:

- 1 Мета, завдання та предмет дисципліни.
- 2 Передумови вивчення дисципліни як основної компоненти освітньо-професійної програми.
- 3 Результати вивчення дисципліни.
- 4 Зміст дисципліни.
- 5 Список рекомендованих джерел.

1. МЕТА, ЗАВДАННЯ ТА ПРЕДМЕТ ДИСЦИПЛІНИ

Метою вивчення навчальної дисципліни «Технології розробки та тестування програмного забезпечення» є засвоєння необхідних знань щодо сучасних методологій, технологій розробки якісних програмних продуктів, а також набуття практичних навичок з використання сучасних методів та засобів тестування програмного забезпечення.

Завданням дисципліни є теоретична та практична підготовка майбутніх фахівців з таких питань: технології розробки програмного забезпечення та їх дефінітивний аналіз; тестування документації та вимог до програмного забезпечення; ручне тестування та його види; техніки тестдизайну; автоматизація тестування програмного забезпечення.

Предметом вивчення дисципліни є сукупність теоретичних і практичних проблем, які пов'язані з технологією розробки програмного забезпечення, методами ручного та автоматизованого тестування.

2. ПЕРЕДУМОВИ ВИВЧЕННЯ ДИСЦИПЛІНИ ЯК ВИБІРКОВОЇ КОМПОНЕНТИ ОСВІТНЬОЇ ПРОГРАМИ

знання: одну мову програмування (наприклад, Python, Java, C++, або іншою), основні концепції та синтаксис; системи контролю версій (наприклад, Git); основи реляційних баз даних, знання SQL.

вміння: програмувати однією з мов програмування; досвід роботи з ОС Windows на рівні адміністрування, писати запити SQL.

3. РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ

Дисципліна «Технології розробки та тестування програмного забезпечення», як обов'язкова компонента освітньої програми, забезпечує оволодіння студентами загальними та фаховими компетентностями і досягнення ними програмних результатів навчання за відповідною освітньо-професійною програмою:

Комп'ютерні науки (ОС магістр)

Номер в освітній програмі	Зміст компетентності	Номер теми, що розкриває зміст компетентності
<i>Загальні компетентності за освітньою програмою</i>		
ЗК2	Здатність застосовувати знання у практичних ситуаціях.	1-13
ЗК5	Здатність вчитися й оволодівати сучасними знаннями.	1-13
<i>Фахові компетентності за освітньою програмою</i>		
СК6	Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук.	3, 6, 7, 8, 9, 10
СК7	Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.	1, 2, 3, 4, 10, 13
СК8	Здатність розробляти і реалізовувати проекти зі створення програмного забезпечення, у тому числі в непередбачуваних умовах, за нечітких вимог та необхідності застосовувати нові стратегічні підходи, використовувати програмні інструменти для організації командної роботи над проектом.	1, 2, 3, 4, 12,
СК10	Здатність оцінювати та забезпечувати якість ІТ проектів, інформаційних та комп'ютерних систем різного призначення, застосовувати міжнародні стандарти оцінки якості програмного забезпечення інформаційних та комп'ютерних систем,	1, 2, 3, 4, 12, 13

	моделі оцінки зрілості процесів розробки інформаційних та комп'ютерних систем.	
СК11	Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.	1-13
<i>Програмні результати навчання за освітньою програмою</i>		
РН11	Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.	1-13
РН13	Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.	1, 2, 3, 4, 5, 12, 13
РН14	Тестувати програмне забезпечення.	1-13
РН15	Виявляти потреби потенційних замовників щодо автоматизації обробки інформації.	1, 2, 3, 4, 5, 10, 12
РН17	Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.	1-13
РН18	Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується	2, 4

4. ЗМІСТ ДИСЦИПЛІНИ

Тема 1. SDLC, SDM. Гнучкі методології розробки програмного забезпечення

Предмет вивчення і задачі дисципліни. Поняття життєвого циклу програмного продукту. Методології розробки програмного забезпечення. Моделі розробки ПЗ: каскадна, еволюційна, покрокова, формальна, спіральна та ін. Стандарти управління життєвим циклом ПЗ (ISO12207, ISO15504). Основи управління якістю розробки. Стандарти серії ISO9000.

Методологія Agile: визначення та коротка історія. Маніфест Agile Software Development. Концепція Scrum методології, основні терміни. Ролі та їх обов'язки в Scrum. Основні практики в Scrum: Daily Scrum Meeting, Sprint Review Meeting, Sprint Abnormal Termination. Артефакти в Scrum: ProductBacklog, SprintBacklog, BurndownChart. Requirements in Scrum. Product Vision. Task, User Story, Epic, Theme.

Список рекомендованих джерел:

Основний: 4 [с. 21-136]

Додатковий: 9 [с. 3-24]

Інтернет-ресурси: 14, 15, 17, 18

Тема 2. Виявлення вимог до продукту. Тестування документації та вимог

Поняття вимог до програмного забезпечення та їх значення. Джерела та шляхи визначення вимог: спільний (мозковий штурм, вимоги до семінарів, фокус-групи, інтерв'ю, опитування/анкетування), дослідження (аналіз документів, аналіз інтерфейсу, моделювання процесів), експерименти (прототипування, спостереження).

Документування вимог до програмної продукції. Документування бізнес-вимог. Документування вимог користувачів та рішення.

Моделювання вимог до продукту. Управління вимогами до продукту.

Рівні та типи вимог. Властивості якісних вимог. Техніки тестування вимог. Приклад аналізу і тестування вимог. Типові помилки під час аналізу і тестування вимог.

Інструментальні засоби для командної роботи під час створення вимог до програмного забезпечення. Trello, як система управління проектами в режимі онлайн в гнучких методологіях розробки програмного забезпечення.

Список рекомендованих джерел:

Основний: 4 [с. 136-182]

Додатковий: 9 [с. 24-36]

Інтернет-ресурси: 14, 15, 17, 18

Тема 3. Поняття та види тестування

Принципи тестування. Підходи до тестування. Схема класифікації тестування. Класифікація за запуском коду на виконання. Класифікація за доступом до коду і архітектурі додатку. Класифікація за ступенем автоматизації. Класифікація за рівнем деталізації додатку (за рівнем тестування). Класифікація за рівнем важливості функцій, які тестуються (за рівнем функціонального тестування). Класифікація за принципами

роботи з додатком. Класифікація за природою додатку. Класифікація за фокусом на рівні архітектури додатку. Класифікація за рівнем залучення кінцевих користувачів. Класифікація за рівнем формалізації. Класифікація за цілями і задачами. Класифікація за техніками і підходами. Класифікація за моментом виконання (хронологією).

Альтернативні та додаткові класифікації тестування. Класифікація за належністю до тестування методом білої та чорної скриньки.

Список рекомендованих джерел:

Основний: 1 [с. 13-67], 2 [с. 18-65]

Додатковий: 8 [с. 3-54]

Інтернет-ресурси: 19, 20, 21

Тема 4. Тестова документація

Поняття Check-list, правила його створення. TestCase і його життєвий цикл. Атрибути (поля) testcase. Інструментальні засоби управління тестуванням. Властивості якісних testcase. Набори testcase. Логіка створення ефективних перевірок. Типові помилки під час розробки check-list, testcase і наборів testcases.

Основна термінологія: помилки, дефекти, збої, відмови тощо. Звіт про дефект та його життєвий цикл. Атрибути (поля) звіту про дефекти. Інструментальні засоби керування звітами про дефекти. Властивості якісних дефектів. Логіка створення ефективних звітів про дефекти. Типові помилки під час написання звітів про дефекти.

Список рекомендованих джерел:

Основний: 4 [с. 200-231]

Додатковий: 10 [с. 8-16]

Інтернет-ресурси: 16, 19, 20, 21

Тема 5. Основні аспекти тест-дизайну, техніки тест-дизайну

Поняття тест-дизайну. Важливість тест-дизайну та його рецензування. Класифікація технік тест-дизайну за Лі Копландом. Техніки тест-дизайну: тестування на основі класів еквівалентності, аналіз граничних значень, таблиця прийняття рішень, метод попарного тестування, тестування на основі причинно-наслідкових зв'язків, тестування на основі сценаріїв, тестування на основі діаграм станів, тестування на основі діаграм потоків даних, тестування на основі вимог.

Список рекомендованих джерел:

Основний: 4 [с. 231-248]

Додатковий: 10 [с. 18-24]

Інтернет-ресурси: 16, 19, 20, 21

Тема 6. Тестування вебдодатків

Архітектура та особливості вебдодатків. Основні види тестування вебдодатків: функціональне тестування; тестування юзабіліті (перевірка зручності користування); тестування сумісності (конфігураційне тестування); тест на продуктивність; перевірка безпеки.

Функціональне тестування вебдодатку: перевірка посилань (внутрішніх, вихідних, ізольованих і т. д.); перевірка форм; тест файлів cookie; валідація HTML/CSS; перевірка бази даних.

Перевірка зручності використання: навігаційне тестування; перевірка контенту; комфорт користування; тестування доступності.

Тестування сумісності: кросплатформений тестінг; перевірка на кросбраузерність; тестування на мобільних пристроях.

Тестування продуктивності: тестування навантаження; стрес-тестування; перевірка на швидкість з'єднання.

Список рекомендованих джерел:

Основний: 4 [с. 248-260]

Додатковий: 10 [с. 24-36]

Інтернет-ресурси: 16, 19, 20, 21

Тема 7. Тестування безпеки вебдодатків

Проєкт по забезпеченню безпеки вебдодатків. Проєкти над якими працює OWASP: OWASP Security Knowledge Framework (SKF), OWASP Mobile Security Testing Guide (MSTG), OWASP Web Security Testing Guide (WSTG), OWASP Zed Attack Proxy (ZAP).

Принципи тестування, методологія тестування за OWASP. Активне та пасивне тестування. Розвідка і уразливості вебдодатків: відкриття вебсторінки/структури програми; збір інформації в вебдодатках; сканування вразливостей вебдодатків.

Список рекомендованих джерел:

Основний: 4 [с. 261-273]

Додатковий: 8 [с. 61-80], 10 [с. 36-41]

Інтернет-ресурси: 20, 21, 22, 23

Тема 8. Автоматизація тестування вебдодатків

Переваги і недоліки автоматизації. Галузь застосування автоматизації. Особливості тест-кейсів в автоматизації. Технології автоматизації тестування.

Використання продуктів Selenium IDE. Selenium WebDriver. Прийоми розробки тестів: пошук елементів (локаторів), дії з елементами, використання фрагментів JavaScript-коду.

Тестування продуктивності. Автоматизація тестів для оцінки продуктивності вебдодатка під різними навантаженнями. Інструменти: JMeter, Gatling, LoadRunner.

Тестування сумісності. Автоматизація тестів для перевірки роботи вебдодатка в різних браузерах та на різних платформах. Інструменти: BrowserStack, Sauce Labs.

API тестування. Автоматизація тестів для перевірки коректності роботи API. Інструменти: Postman, REST Assured, SoapUI.

Інфраструктура запуску тестів: запуск з консолі, запуск в різних браузерах, зберігання тестів на прикладі сервісу GitHub. Керування потоками виконання тестів.

Список рекомендованих джерел:

Основний: 4 [с. 273-281]

Додатковий: 7 [с. 3-320]

Інтернет-ресурси: 11, 12, 13

Тема 9. Тестування мобільних додатків

Виникнення мобільних додатків. Різновиди мобільних додатків. Особливості тестування мобільних додатків. Типи тестування мобільних додатків: тестування зручності використання, тестування сумісності, тестування інтерфейсу, тестування сервісів, тестування ресурсів низького рівня, тестування продуктивності, тестування експлуатації, тестування встановлення, тестування безпеки.

Стратегія тестування мобільних додатків. Техніки тестування безпеки мобільних додатків. Автентифікація, авторизація, безпека даних, уразливості для злому, управління сесіями.

Інструменти тестування мобільних додатків: ImmuniWeb MobileSuite; Zed Attack Proxy; Kiuwan; QARK; Micro Focus; Android Debug Bridge; CodifiedSecurity; Drozer; WhiteHat Security; Synopsys; Veracode; Mobile Security Framework.

Список рекомендованих джерел:

Основний: 2 [152-167]

Додатковий: 9 [с. 80-90], 10 [с. 81-92]

Інтернет-ресурси: 16, 18, 19

Тема 10. Автоматизація тестування мобільних додатків

Знайомство з інструментами розробки та тестування мобільних додатків. Інструменти для автоматизації та їх дефінітивний аналіз. Проблеми та особливості Appium та WebDriver. Середовище розробки. Запуск та налаштування Appium. Знайомство з

UIAutomatorViewer. Appium Inspector. Підключення до програми, пошук елементів інтерфейсу, використання в кодї.

Локатори: Xpath, Id, Class, Name, Css (mobile web only).

Підготовка тестового додатку, підготовка сценарію тест-кейсу, застосування вбудованих методів Click, Find_element, Send_keys та ін. Очікування Implicity vs Explicity, валідація та Asserts.

Паттерни та антипатерни. Тестування жестів Swipe, Reporting, Listeners. Особливості автоматизації тестування мобільних вебдодатків.

Список рекомендованих джерел:

Основний: 1 [с. 172-256], 4 [с. 280-296]

Додатковий: 5 [с. 4-240]

Інтернет-ресурси: 11, 12, 13, 16, 18, 19

Тема 11. Тестування комп'ютерних ігор

Геймдизайн та його тестування. Основні аспекти тестування геймдизайну: ігровий баланс, інтуїтивність геймплея, реакція гравців.

Ігрові механізми, тестування ігрових механізмів: контроль і управління, бойові та інші системи, система винагород.

Тестування продуктивності та тестування FPS.

Тестування сумісності: тестування кросплатформенності, конфігураційне тестування. Тестування Геймплея. Тестування рівнів: перевірка складності, тестування логіки рівнів.

Тестування сценаріїв: тестування сюжетних ліній, перевірка інтерактивності. Тестування користувацького інтерфейсу (UI).

Особливості тестування ігрових платформ. Вимоги до сертифікації: дотримання вимог сертифікації кожної платформи (правила PlayStation Certification).

Тестування ігор на мобільних пристроях.

Інструменти для тестування комп'ютерних ігор: Unity Test Tools, Unreal Engine Automation Testing, Appium, Selenium, Fraps.

Список рекомендованих джерел:

Основний: 1 [с. 256-308], 4 [с. 296-430]

Додатковий: 6 [с. 126-153], 7 [23-40]

Інтернет-ресурси: 11, 12, 13, 16, 18, 19

Тема 12. ШІ в управлінні проєктами та тестуванні програмного забезпечення

Управління проєктами за допомогою ШІ. Оптимізація розподілу ресурсів. Аналіз ризиків. Підтримка прийняття рішень. Автоматизація управлінських процесів на основі ШІ.

Автоматизація тестування на основі ШІ: автоматизація виконання тестових сценаріїв, інструменти ШІ для автоматизації тестування. Генерація тестових випадків та даних на основі роботи ШІ. Аналіз результатів тестування на основі ШІ. Прогнозування дефектів. Оптимізація тестового покриття: аналіз програмного коду для визначення областей, які потребують тестування; інструменти ШІ для оптимізації тестового покриття.

Інтеграція ШІ в процеси управління та тестування. Інтеграція в процес безперервної інтеграції/розгортання (CI/CD). Моніторинг та звітність.

Виклики та перспективи використання ШІ. Необхідність великих обсягів даних. Етичні питання та приватність. Підтримка та навчання моделей.

Список рекомендованих джерел:

Основний: 4 [с. 438-521]

Додатковий: 8 [с. 90-94]

Інтернет-ресурси: 24, 25

Тема 13. Метрики та методики оцінки. Оцінка трудовитрат, планування та звітність в тестуванні

Метрики якості. Метрики за тестовими випадками. Метрики за багами/дефектами. Метрики за задачами. Юзабіліті-метрики. Метрики Kanban. Метрики SCRUM. Валідація і верифікація.

Планування і звітність. Тест-план і звіт про результати тестування. Оцінка трудовитрат.

Список рекомендованих джерел:

Основний: 4 [с. 613-682]

Додатковий: 10 [с. 23-41]

Інтернет-ресурси: 11, 14, 16, 20, 21

5. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Jorgensen P. C., DeVries B. Software testing: a craftsman's approach, fifth edition. Auerbach Publishers, Incorporated, 2021. 528 p.
2. Jorgensen P., Vries B. D. Software testing. Taylor & Francis Group, 2021. 423p.
3. Niranjnamurthy M., Khaiyum S., Rakshitha K. P. Trends in software testing. Wiley & Sons, Incorporated, John, 2022. 341 p.
4. Ramonyai J. Software engineering: software development, architecture, design patterns, testing, product management, project lifecycle, TR5,

programming, quality assurance, security systems, service and practic. Independently Published, 2022. 729 p.

Додатковий

5. Chandrasekara C. Hands-On Functional Test Automation: With Visual Studio 2017 and Selenium Paperback / Chaminda Chandrasekara, Pushpa Herath—Apress: May 2, 2019. – 252 p.
6. Gundecha M. Selenium Testing Tools Cookbook –Second Edition 2nd Edition / Unmesh Gundecha – Packt Publishing : November 23, 2019. – 326 p.
7. Horalek J., Urbanik P., Sobeslav V. Automated tests using selenium framework. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Cham, 2023. P. 23–40.
8. Вакалюк Т.А. Технології тестування програм. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика. – Житомир: Вид-во ЖДУ, 2018. – 196 с.
9. Карпенко М. Ю. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.
10. Золотухіна О.А. Якість та тестування інформаційних систем. Навчальний посібник підготовлено до друку для самостійної роботи студентів вищих навчальних закладів. Золотухіна О.А., Негоденко О.В., Резник С.Ю., Разіна С.Я. Київ: ННІТ ДУТ, 2020. –128 с.

Інтернет-ресурси

11. Automated testing. Encyclopedia of computer graphics and games. Cham, 2024. P. 211. URL: https://doi.org/10.1007/978-3-031-23161-2_300154 (date of access: 10.06.2024).
12. Implementing Automated Software Testing – Continuously Track Progress and Adjust Accordingly – URL: <http://www.methodsandtools.com/archive/archive.php?id=94> (date of access: 22.05.24p.)
13. Parsa S. Software testing automation. Cham : Springer International Publishing, 2023. URL: <https://doi.org/10.1007/978-3-031-22057-9> (date of access: 10.06.2024).
14. The Agile System Development Life Cycle (SDLC) – URL: <http://www.ambysoft.com/essays/agileLifecycle.html> (date of access: 24.05.24p.)

15. The History of Software Testing– URL: <http://www.testingreferences.com/testinghistory.php> (date of access: 20.05.24p.)
16. What is documentation testing in software testing?– URL: <http://tryqa.com/what-is-documentation-testing/> (date of access: 22.05.24p.)
17. What is V-model - advantages, disadvantages and when to use it?– URL: <http://tryqa.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/> (date of access: 10.06.24p.)
18. Моделювання програмного забезпечення : навч.-метод. посіб. [Електронний ресурс] / С. Ю. Манаков та ін. Одеса : Фенікс, 2023. URL: <https://doi.org/10.32837/11300.25952> (дата звернення: 10.06.2024).
19. Родіонов П., Полупан Ю., Родіонова О. Теоретичні засади тестування програмного забезпечення. Наука і техніка сьогодні. 2024. № 3(31). URL: [https://doi.org/10.52058/2786-6025-2024-3\(31\)-941-952](https://doi.org/10.52058/2786-6025-2024-3(31)-941-952) (дата звернення: 10.06.2024).
20. O'Regan G. Software quality assurance. Undergraduate topics in computer science. Cham, 2022. P. 239–246. URL: https://doi.org/10.1007/978-3-031-07816-3_13 (date of access: 10.06.2024).
21. Технічні статті | Онлайн-курси від компанії QATestLab. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/> (дата звернення: 10.06.2024).
22. Certified Tester Foundation Level Syllabus. International Software Testing Qualifications Board. URL: <https://www.istqb.org/> (date of access: 10.06.2024).
23. OWASP web security testing guide | OWASP foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: <https://owasp.org/www-project-web-security-testing-guide/> (date of access: 10.06.2024).
24. Штучний інтелект у тестуванні ігор. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/artificial-intelligence-in-game-testing/> (дата звернення: 10.06.2024).
25. Is AI the ultimate QA? / E. M. Weisberg et al. Journal of digital imaging. 2022. URL: <https://doi.org/10.1007/s10278-022-00598-8> (date of access: 10.06.2024).

**Курсивом зазначені джерела, що є в наявності в бібліотеці ДТЕУ*