

Чернівецький національний університет імені Юрія Федьковича
(повне найменування закладу вищої освіти)

Факультет математики та інформатики
(назва факультету/навчально-наукового інституту)

Кафедра математичного моделювання
(назва кафедри, що забезпечує викладання)

“ЗАТВЕРДЖУЮ”

**Декан факультету
математики та інформатики**

Ольга МАРТИНЮК



“25”

2025 року

**РОБОЧА ПРОГРАМА
навчальної дисципліни**

Програмування та підтримка вебзастосунків

(назва навчальної дисципліни)

вибіркова

(вказати: обов'язкова)

Освітньо-професійна програма «Системний аналіз»

(назва програми)

Спеціальність 124 Системний аналіз

(вказати: код, назва)

Галузь знань 12 Інформаційні технології

(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)

(вказати: перший (бакалаврський) / другий (магістерський) / третій (освітньо-науковий))

Факультет математики та інформатики

(назва факультету/ навчально-наукового інституту, на якому здійснюється підготовка фахівців за вказаною освітньою програмою)

Мова навчання українська

(вказати: на якій мові читається дисципліна)

Чернівці 2025 рік

Робоча програма навчальної дисципліни «Програмування та підтримка вебзастосунків» складена відповідно до освітньо-професійної програми «Системний аналіз»

Розробник:

Дорош Андрій Богданович, асистент, кандидат фіз.-мат. наук

Викладач, що забезпечує читання даної навчальної дисципліни:

Дорош Андрій Богданович, асистент, кандидат фіз.-мат. наук

Затверджено на засіданні кафедри математичного моделювання

Протокол № 15 від «24» червня 2025 року

Завідувач кафедри _____

Ігор ЧЕРЕВКО

Схвалено методичною радою факультету математики та інформатики

Протокол № 12 від «25» червня 2025 року

Голова методичної ради _____

Віра СІКОРА

Мета навчальної дисципліни

Мета вивчення навчальної дисципліни – оволодіти основними принципами проектування веб-сайтів та веб-застосунків, освоїти скриптову серверну мову програмування PHP, основи налаштування необхідного програмного забезпечення на веб-сервері.

Завдання вивчення навчальної дисципліни

Після вивчення даної дисципліни студент повинен знати:

- принципи проектування веб-застосунків;
- основні можливості мови програмування PHP;
- принципи роботи з СКБД MySQL;
- можливості системи контролю версій Git.

Після вивчення даної дисципліни студент повинен вміти:

- проектувати структуру файлової системи веб-застосунків;
- застосовувати мови PHP для обробки запитів на сервері;
- будувати запити до бази даних MySQL;
- фіксувати зміни в проекті за допомогою Git.

Пререквізити

Для вивчення даної освітньої компоненти будуть базою такі дисципліни: Програмування, Об'єктно-орієнтоване програмування.

Результати навчання:

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

- ЗК01. Здатність до абстрактного мислення, аналізу та синтезу
- ЗК02. Здатність застосовувати знання у практичних ситуаціях
- ЗК04. Знання та розуміння предметної області та розуміння професійної діяльності
- ЗК05. Здатність спілкуватися державною мовою усно і письмово
- ЗК06. Здатність спілкуватися іноземною мовою
- ЗК07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел
- ЗК10. Здатність працювати автономно
- ЗК14. Здатність оцінювати та забезпечувати якість виконуваних робіт

фахові компетентності:

ФК5. Здатність формулювати задачі оптимізації при проектуванні систем управління та прийняття рішень, а саме: математичні моделі, критерії оптимальності, обмеження, цілі управління; обирати раціональні методи та алгоритми розв'язання задач оптимізації та оптимального керування.

ФК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

ФК8. Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.

програмні результати навчання:

ПР8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.

ПР11. Знати і вміти застосовувати на практиці системи управління базами даних і знань та інформаційні системи.

ПР13. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах

Опис змісту робочої програми навчальної дисципліни
Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	3	5	3	90	15	–	–	30	45	–	залік

Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма							заочна форма					
	усього	у тому числі					усього	у тому числі					
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	
Змістовий модуль 1													
Тема 1. Основи мови програмування PHP	16	4		6		6							
Тема 2. Встановлення веб-сервера	10	2		4		4							
Тема 3. Створення	7	1		2		4							

<i>єдиної точки входу</i>													
<i>Тема 4. GET і POST-запити. Реєстрація та авторизація користувача</i>	9	1		4		4							
Разом за ЗМ 1	42	8		16		18							
Змістовий модуль 2													
<i>Тема 5. Валідація даних, надісланих з форми. Регулярні вирази</i>	12	2		4		6							
<i>Тема 6. Робота з базою даних</i>	9	1		2		6							
<i>Тема 7. Сесія користувача та файли cookies</i>	12	2		4		6							
<i>Тема 8. ООП на PHP</i>	15	2		4		9							
Разом за ЗМ 2	48	7		14		27							
Усього годин	90	15		30		45							

Теми лекційних занять

№ з/п	Назва теми
1	Вступ. Історія розвитку мови програмування PHP.
2	Встановлення веб-сервера.
3	Призначення PHP. Розбиття HTML на файли.
4	Змінні, розгалуження, цикли.
5	Масиви.
6	GET-запити.
7	POST-запити.
8	Регулярні вирази.
9	Робота з базами даних.
10	Файли cookies.
11	Сесія користувача.
12	Створення функцій.
13	Класи та об'єкти. Чарівні методи.
14	Успадкування класів.
15	Абстрактні класи та інтерфейси.

Теми лабораторних занять

№ з/п	Назва теми (завдання)	Кількість годин
1	<p>Тема: Створення базової файлової структури сайту.</p> <p>Завдання: Обрати чітку тематику для свого сайту. Встановити веб-сервер Apache. Його можна завантажити окремо або у складі пакетів програм XAMPP, WAMP, Denwer чи OpenServer. Створити веб-сторінку засобами HTML та CSS із чіткою структурою - заголовочна частина, бокове меню (зліва, справа чи обидва) та нижня частина. У меню мають бути два пункти: "Головна" і "Про сайт". "Розрізати" веб-сторінку на окремі файли: header.php, left_menu.php, right_menu.php (якщо є), footer.php. Помістити у відповідні файли відповідні частини HTML-коду. У файлі index.php має залишитись лише контент (вміст) сторінки. Крім того, у файлі index.php має бути PHP-код, який підключає вміст інших файлів. Використати систему контролю версій git.</p>	4
2	<p>Тема: Створення єдиної точки входу.</p> <p>Завдання: У кореневій директорії створити підпапки js, css та img, куди будуть поміщатись відповідні файли. Створити підкаталог layout, помістити в нього усі PHP-скрипти, створені раніше, окрім index.php. Створити підкаталог views і в ньому два нові файли: main.php та about.php. У перший файл помістити контент головної сторінки сайту, який раніше знаходився у файлі index.php. Під контентом маєтсья на увазі лише HTML-код із текстом сторінки. А PHP-код, який відповідає за підключення хедера та футера, має залишитись у файлі index.php. У файл about.php помістити HTML-код із коротким описом сайту та його автора. Відтепер доступ до всіх сторінок сайту здійснюється через точку входу - файл index.php, залежно від того, чому дорівнює параметр action у суперглобальному масиві \$_GET. Якщо цей параметр відсутній або невірний, то показувати сторінку main.php. Наприклад, за адресою index.php?action=about повинна відображатись сторінка about.php, за адресою index.php - сторінка main.php, а за неіснуючою адресою index.php?action=123 - також сторінка main.php. У меню сайту пункти "Головна" і "Про сайт" тепер повинні вказувати на відповідні сторінки. Перевірка параметру action у масиві \$_GET не повинна виглядати як перелік усіх наявних на сайті сторінок і має займати лише один блок if.</p>	4
3	<p>Тема: Створення сторінки реєстрації користувача.</p> <p>Завдання:</p>	4

	<p>У директорії views створити файл registration.php (на сайті вона буде відкриватися через index.php?action=registration) із HTML-кодом форми реєстрації нового користувача. Там мають бути наступні поля:</p> <ul style="list-style-type: none"> • Логін • Пароль • Повторіть пароль • Електронна пошта • Ще одне поле, згідно свого варіанту завдання • Кнопка "Зареєструватися" <p>Кожне поле повинне мати унікальний атрибут name. При створенні форми використати теги <label>.</p> <p>Коли користувач натисне кнопку "Зареєструватися", скрипт має здійснити перевірку, чи правильно заповнені усі поля. Обов'язково використати регулярні вирази.</p> <p>Якщо якийсь із полів заповнене невірно, має відкритися ця сама сторінка, але з повідомленням про те, які саме поля були невірно заповнені і чому. Якщо ж усі поля заповнені вірно, користувач має побачити нову сторінку (наприклад, index.php?action=registration_successful) із повідомленням про те, що реєстрація пройшла успішно (форму реєстрації при цьому вже не показувати), а нижче - посилання "На головну", яке відкриває головну сторінку сайту.</p>	
4	<p>Тема: Підключення до бази даних MySQL.</p> <p>Завдання:</p> <p>Створити базу даних MySQL. Її назва має відображати тематику сайту. Це можна зробити у графічному веб-середовищі phpmyadmin. Воно постачається у складі пакету XAMPP, але за потреби можна завантажити окремо тут.</p> <p>У новоствореній базі створити таблицю users. У ній має бути обов'язкове поле id (або user_id, id_user) - первинний ключ, а також усі інші поля, описані у відповідному варіанті попередньої лабораторної роботи. Типи цих полів повинні відповідати типу значень, яке вони зберігатимуть (ціле число, текст тощо).</p> <p>Вставити у скрипті реєстрації код, який додає нового користувача до таблиці users, якщо всі поля заповнені вірно.</p> <p>Пароль користувача має зберігатись у базі даних у захешованому вигляді за допомогою алгоритму Blowfish. Для цього використати функцію password_hash.</p>	4
5	<p>Тема: Авторизація користувачів.</p> <p>Завдання:</p> <p>Додати до таблиці users у базі даних поле admin типу boolean (хоча такий тип є у списку, насправді він зберігатиметься як tinyint(1), оскільки в MySQL немає окремого типу boolean, це лише псевдонім). Створити двох користувачів: адміністратора (вручну задати полю admin значення 1 у БД) та звичайного користувача (зі значенням 0 у полі admin).</p> <p>У директорії views створити файл login.php із HTML-кодом форми авторизації користувача. Там мають бути наступні поля:</p> <ul style="list-style-type: none"> • Логін • Пароль 	4

	<ul style="list-style-type: none"> • Кнопка "Увійти". <p>Коли користувач натисне кнопку "Увійти", скрипт має здійснити перевірку, чи є користувач із вказаним логіном у базі даних. Якщо є, то чи співпадає хеш введеного паролю із тим хешем, що у базі (для цього використати функцію <code>password_verify</code>). Якщо не виконується будь-яка з вищевказаних умов, вивести знову форму авторизації із повідомленням "Невірний логін або пароль". Якщо ж усі дані введені вірно, записати до сесії логін, ID користувача та ознаку, чи він є адміністратором, і перенаправити на головну сторінку сайту (або просто показати посилання на головну).</p> <p>Додати до меню сайту два нові пункти:</p> <ul style="list-style-type: none"> • Увійти - цей пункт мають бачити лише неавторизовані користувачі (гості) • Вийти - цей пункт мають бачити лише авторизовані користувачі. Це посилання має вести на нову сторінку <code>logout</code>. Ця сторінка видаляє із сесії всі дані про поточного користувача і перенаправлює на головну сторінку (або показує посилання на головну). 	
6	<p>Тема: Створення CRUD-сторінок.</p> <p>Завдання:</p> <p>Створити нову таблицю у базі даних і назвати її згідно тематики свого сайту (наприклад, <code>automobiles</code>). Додати поля, які характеризують автомобіль. Наприклад, марка, модель, рік випуску тощо.</p> <p>Створити сторінку з формою додавання інформації про новий автомобіль (<code>index.php?action=create_automobile</code>). Здійснювати перевірку, чи поля заповнені вірно. Якщо так, то додати запис до таблиці <code>automobiles</code>. Додати до меню сайту пункт "Додати автомобіль". Цей пункт можуть бачити лише авторизовані користувачі. Якщо автомобіль додає адміністратор сайту, то новий запис у таблиці матиме значення поля <code>visible</code> 1. Якщо ж не адміністратор, то 0.</p> <p>Створити сторінку зі списком автомобілів (<code>index.php?action=automobiles</code>), а також додати до меню пункт "Автомобілі" (видимий для всіх користувачів). На цій сторінці буде показано інформацію про всі автомобілі з таблиці <code>automobiles</code>, у яких значення поля <code>visible</code> дорівнює 1 (але адміністратор повинен бачити всі автомобілі, разом із неопублікованими). Новіші записи мають бути на початку. Список можна оформити довільним чином - у вигляді таблиці або ж окремих абзаців.</p> <p>Створити сторінку редагування існуючого автомобіля (<code>index.php?action=update_automobile&id=5</code>), доступну лише для адміністратора. Скрипт повинен зчитувати параметр <code>id</code> з адресного рядка браузера. Наприклад, <code>id=5</code> вказує, що потрібно витягнути з БД із таблиці <code>automobiles</code> запис, у якого значення поля <code>id</code> (або <code>id_automobile</code>) дорівнює 5. Якщо у базі даних не існує автомобіля з таким ID, показувати повідомлення, що такої сторінки не існує. Дана сторінка може мати такий же вигляд, як і сторінка додавання нового автомобіля, аналогічно перевіряючи, чи коректно заповнені усі поля форми. Єдина відмінність - коли</p>	4

	<p>користувач відкриває сторінку редагування 5-го автомобіля, текстові поля уже мають бути заповнені інформацією про цей автомобіль. Крім того, дати можливість обирати, чи видимий він на сайті (редагування поля visible із бази даних).</p> <p>Створити сторінку перегляду існуючого автомобіля (index.php?action=view_automobile&id=5), доступну для всіх користувачів. Відмінність від пункту 3 полягає в тому, що на даній сторінці буде лише інформація про один автомобіль, а не всі зразу. Якщо у базі даних не існує автомобіля з таким ID, показати повідомлення, що такої сторінки не існує.</p> <p>Створити сторінку видалення існуючого автомобіля (index.php?action=delete_automobile&id=5), доступну лише для адміністратора. Якщо у базі даних не існує автомобіля з таким ID, показати повідомлення, що такої сторінки не існує. Якщо ж існує, то видалити вказаний автомобіль і показати повідомлення, що автомобіль успішно видалено.</p> <p>Сторінки, створені у попередніх пунктах, поки що ніяк не можна переглянути, адже на сайті ніде немає посилання на них. Для цього на сторінці зі списком автомобілів біля кожного автомобіля слід додати кнопки: "Перегляд", "Редагувати", "Видалити". Останні дві буде бачити лише адміністратор. Ці кнопки відкриватимуть відповідні сторінки, але з різним параметром id. Наприклад, якщо у базі є два автомобілі з ID 1 та 8, то кнопка "Перегляд" біля першого автомобіля відкриватиме сторінку index.php?action=view_automobile&id=1, а біля другого - сторінку index.php?action=view_automobile&id=8. При натисненні кнопки "Видалити" спочатку запитувати, чи дійсно користувач хоче видалити інформацію про вказаний автомобіль.</p>	
7	<p>Тема: Додатковий функціонал на сайті.</p> <p>Завдання:</p> <p>Створити посторінкову навігацію статей/записів. Показувати по 20 на сторінці. Скажімо, у БД є 48 автомобілів. Тоді на першій сторінці будуть перші 20. Знизу будуть кнопки Перша, 1,2,3, Остання. Причому Перша і 1 не будуть клікабельними, оскільки відкрита перша сторінка. Коли користувач натисне 2, то відкриються наступні 20 записів і кнопка 2 не буде клікабельною. На третій сторінці будуть останні 8 автомобілів. Номер сторінки передавати через додатковий параметр page в адресі сторінки, наприклад, index.php?action=automobiles&page=2. Підказка: щоб витягати лише по 20 записів із БД, потрібно в SQL-запиті використати ключове слово LIMIT.</p>	4
8	<p>Тема: Розгортання сайту на зовнішньому сервері.</p> <p>Завдання:</p> <p>Усі файли проекту, зроблені в ході попередніх лабораторних робіт, потрібно заархівувати і закинути на зовнішній сервер через файловий менеджер у браузері. Логін та пароль до сервера надаються після реєстрації. За бажанням, можна використовувати FTP.</p> <p>Базу даних потрібно окремо експортувати з локального комп'ютера. Для цього у середовищі phpmyadmin слід вибрати</p>	2

	свою базу даних і натиснути вкладку Експортувати. Потім зберегти її у файл із розширенням .sql. У панелі керування хостингу також має бути середовище phpmyadmin, у якому треба імпортувати збережений sql-файл.	
	Разом	30

Самостійна робота

Самостійна робота складає 45 годин. Розподіл самостійної роботи за видами навчальних робіт:

- підготовка до лекційних занять – 10 годин;
- виконання лабораторних робіт – 30 годин;
- підготовка до заліку – 5 годин.

№ з/п	Назва теми	Кількість годин
1	Підготовка до лекційних занять.	6
2	Встановлення веб-сервера.	4
3	Створення єдиної точки входу.	4
4	Система контролю версій git.	4
5	Створення CRUD-сторінок.	6
6	Створення додаткового функціоналу на сайті.	6
7	Розгортання веб-сайту на віддаленому сервері.	6
8	Підготовка до заліку.	9
	Разом	45

Контроль та оцінювання навчальних досягнень студентів у процесі вивчення навчальної дисципліни

Під час вивчення навчальної дисципліни використовуються такі види та методи контролю: *лабораторні роботи та тести.*

Перелік питань для самоконтролю й контролю навчальних досягнень студентів з навчальної дисципліни:

- 1) Що таке вебсервер?
- 2) Що таке HTTP-запит?
- 3) Яке призначення мови програмування PHP?
- 4) Де виконуються PHP-скрипти?
- 5) На яких операційних системах можуть виконуватись PHP-скрипти?
- 6) Чи можуть бачити користувачі PHP-код вебсайту?
- 7) Які основні типи даних у мові PHP?
- 8) Як оголошуються змінні у PHP?
- 9) Чи назви змінних чутливі до регістру?
- 10) Як здійснюється перетворення типів даних?

- 11) Які є типи циклів?
- 12) Які бувають типи масивів у PHP?
- 13) Якими можуть бути ключі (індекси) елементів масивів?
- 14) Які є способи перебрати всі елементи масиву?
- 15) Як вивести всі елементи масиву на екран без використання циклів?
- 16) Як передавати в цикл оригінал або копію кожного елемента масиву?
- 17) Що таке регулярний вираз? Як перевіряти у PHP?
- 18) Як здійснюється обробка даних, введених користувачем з HTML-форми?
- 19) Як захешувати пароль?
- 20) Яка різниця між GET і POST-запитами?
- 21) Як описуються класи, методи та поля у PHP?
- 22) Які бувають модифікатори доступу до полів та методів класу?
- 23) Що таке статичне поле класу?
- 24) Що таке статичний метод класу?
- 25) Що таке чарівні методи?
- 26) Що таке абстрактний клас?
- 27) Чим відрізняється інтерфейс від абстрактного класу?

Критерії оцінювання навчальних досягнень студентів за результатами поточного контролю

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS та є накопичувальною.

Протягом семестру студенти виконують 8 лабораторних робіт та 2 контрольних тестування. Лабораторні роботи (ЛР) оцінюються різною кількістю балів, контрольні тестування по 10 балів кожне, всього протягом семестру можна набрати 70 балів (див. таблицю нижче).

Поточне оцінювання (<i>аудиторна та самостійна робота</i>)										Кількість балів (залік)	Сумарна к-ть балів
ЛР1	ЛР2	ЛР3	ЛР4	Тестування	ЛР 5	ЛР 6	ЛР 7	ЛР 8	Тестування	30	100
10	10	5	5	10	5	5	5	5	10		

Зарахування результатів неформальної освіти

Здобувачі вищої освіти має право на участь у неформальній/інформальній освіті.

У межах поточного контролю можуть визнаватися результати неформальної/інформальної освіти за умови наявності сертифікату або освітньої декларації про результати неформальної/інформальної освіти з питань, що відповідає тематиці курсу («Порядок визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти», <https://www.chnu.edu.ua/media/4g5fzssb/poriadok-vyznannia-rezultatuv-navchannia-zdobutykh-shliakhom-neformalnoi-ta-abo-informalnoi-osvity.pdf>).

Студентам можуть бути зараховані додаткові бали, отримані через неформальну освіту, до загальної суми балів, набраної з освітньої компоненти, за умови, що результати з проблеми, за якою відбувалося навчання, відповідають тематиці курсу.

Критерії оцінювання результатів навчання (з навчальної дисципліни) на підсумковому контролі

Критерієм підсумкового оцінювання є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним запланованим результатом навчання навчальної дисципліни.

Протягом вивчення дисципліни студент за виконання лабораторних робіт повинен набрати не менше 50% можливих балів (25 балів з 50) та на підсумковому тестуванні набрати не менше 50% можливих балів (10 балів з 20).

Підсумкова оцінка, як показник результатів вивчення навчальної дисципліни, складається із сумарної кількості балів за поточне оцінювання – **70 балів** та підсумкового модуль-контролю (залік) – **30 балів**, за **100-бальною університетською шкалою**.

Шкала оцінювання: національна та ЄКТС

100-бальна шкала	Оцінка за національною шкалою		Оцінка за шкалою ЄКТС	
			Оцінка	Пояснення за розширеною шкалою
90-100	Зараховано	Відмінно	A	відмінно
80-89		Добре	B	дуже добре
70-79			C	добре
60-69		Задовільно	D	задовільно
50-59			E	достатньо
35-49	Незараховано	Незадовільно	FX	(незадовільно) з можливістю повторного складання
1-34			F	(незадовільно) з обов'язковим самостійним повторним опрацюванням освітнього компонента до перескладання

Методи та освітні технології навчання

Лекції, лабораторні роботи, тестування, аудиторне та онлайн-навчання з використанням систем Moodle та Google Meet.

Методи навчання:

- вербальні методи (лекція, бесіда, диспут, пояснення, розповідь тощо);
- практичні методи (лабораторні роботи);
- наочні методи (демонстрація, ілюстрація);
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою та інтернет-ресурсами;
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни;
- навчання з використанням відповідних онлайн-платформ.

Рекомендована література

Основна

1. Дорош А. Б., Іліка С. А. Програмування та підтримка вебзастосунків. Навчально-методичний посібник. Чернівецький національний університет імені Юрія Федьковича. — Чернівці, 2025. — 120 с.
2. David Powers. PHP 7 Solutions: Dynamic Web Design Made Easy. 4th Edition. – Apress, 2019. – 580 p.
3. Josh Lockhart. Modern PHP: New Features and Good Practices. – O'Reilly Media, 2015. – 270 p.

Додаткова

1. Robin Nixon. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. – O'Reilly Media, 2018. – 832 p.
2. Matt Zandstra. PHP Objects, Patterns, and Practice. 5th Edition. – Apress, 2016. – 608 p.

8. Інформаційні ресурси

1. Офіційний сайт PHP: <https://www.php.net>
2. Довідковий сайт про PHP: <https://www.w3schools.com/php/default.asp>
3. Електронний курс у системі Moodle:
<https://moodle.chnu.edu.ua/course/view.php?id=194>
4. Програмування та підтримка вебзастосунків. Навчально-методичний посібник: <https://archer.chnu.edu.ua/xmlui/handle/123456789/12624>

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

1. «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» <https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/etychnyi-kodeks-chernivetsko-ho-natsionalnoho-universytetu-imeni-yurii-a-fedkovycha/>

2. «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Ча» <https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu/>