

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Робоча програма
навчальної дисципліни
"WEB-ПРОГРАМУВАННЯ"
для студентів напряму підготовки "Комп'ютерні науки"
денної форми навчання

Харків. Вид. ХНЕУ, 2009

Затверджено на засіданні кафедри інформаційних систем.
Протокол №5 від 12.12.2008 р.

P78 Робоча програма навчальної дисципліни "Web-програмування" для студентів напряму підготовки "Комп'ютерні науки" денної форми навчання / Укл. О. В. Щербаков, В. В. Огурцов. — Харків: Вид. ХНЕУ, 2009. — 28 с. (Укр. мов.)

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами, вміщено плани лекцій і лабораторних занять, матеріал щодо закріплення знань (індивідуальне завдання, самостійна робота, контрольні запитання), методичні рекомендації та оцінювання знань студентів.

Рекомендовано для студентів напряму підготовки "Комп'ютерні науки".

Вступ

Нова інформаційна технологія досягла такого розвитку, що, мабуть, не залишилося сфер людського життя, які незачеплені глобальною мережею Internet. У цей час інтерес до мережі Internet продовжує зростати. Розроблений у роки інформаційного вибуху Internet стає невід'ємною частиною життя більшості людей усього світу.

Якісний сайт стає важливим, а у деяких галузях – єдиним, засобом досягнення економічних, політичних, соціальних, рекламних та інших цілей. Якісний сайт відрізняється від інших сайтів у глобальній мережі (яких більшість) такими рисами: висока якість інформаційного наповнення й грамотність його подачі; оригінальність і естетична привабливість зовнішнього вигляду сторінок; доступність змісту сайту для максимально широкого кола користувачів поза залежністю від застосовуваних ними типів пристроїв і версій браузерів, а також від обмежень за станом здоров'я; ергономічність елементів користувальницького інтерфейсу сайту, що забезпечує високу ефективність, але в той же час легкість і невимушеність взаємодії відвідувача з веб-ресурсом; надійність і безпека використовуваних технологічних рішень, чітка погодженість роботи всіх компонентів; бездоганне пророблення всіх деталей і нюансів.

Але для створення якісного сайту, тобто відповідності цим рисам, потрібна плідна робота висококваліфікованих спеціалістів із різних Web-технологій, які повинні розуміти не лише вузьку галузь знань але і добре представляти весь спектр Web-технологій.

Метою навчальної дисципліни є засвоєння необхідних знань з основ Web-програмування, а також формування твердих практичних навичок щодо розробки якісних сайтів.

Предметом вивчення дисципліни є принципи Web-програмування, а також методи їх використання при розробці сайтів різноманітного призначення.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з технічною літературою, та сучасним програмними засобами розробки програм.

Структура робочої програми навчальної дисципліни "Web-програмування" наведена в табл. 1.

Структура програми навчальної дисципліни

Навчальна дисципліна: підготовка бакалаврів	Напрямок, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів, відповідних до ECTS – 3; у тому числі: змістовних модулів – 2; самостійна робота; індивідуальне завдання	Шифр та назва напрямку: 6.050101 "Комп'ютерні науки"	Вибіркова. Рік підготовки: 3. Семестр: 6
Кількість годин: усього – 108; за змістовними модулями: модуль 1 – 52 години; модуль 2 – 56 години		Лекції: кількість годин – 16. Лабораторні заняття: кількість годин – 32. Самостійна робота: кількість годин – 44. Індивідуальна робота: кількість годин – 16
Кількість тижнів викладання дисципліни: 16. Кількість годин на тиждень – 3	Освітньо-кваліфікаційний рівень: бакалавр	Вид контролю: іспит

У процесі навчання студенти отримують необхідні знання під час проведення аудиторних занять: лекційних та лабораторних. Велике значення в процесі вивчення та закріплення знань має самостійна робота студентів.

Зміст усіх видів занять розроблено відповідно до кредитно-модульної системи організації навчального процесу.

1. Кваліфікаційні вимоги до студентів у галузі комп'ютерних наук

Дисципліна "Web-програмування" є базовою для підготовки бакалаврів напрямку підготовки "Комп'ютерні науки".

Необхідна навчальна база перед початком вивчення дисципліни

З метою найкращого засвоєння матеріалу студенти повинні до початку вивчення дисципліни засвоїти теоретичні знання та опанувати практичні вміння з дисципліни "Основи програмування та алгоритмічні мови", а також мати навички роботи з персональним комп'ютером.

У результаті вивчення даної дисципліни студенти повинні:

знати:

- правила побудови документів HTML;
- основні властивості каскадних таблиць стилів;
- основні елементи об'єктної моделі браузера;
- правила побудови і основні елементи сценаріїв мовою JavaScript;
- правила побудови і основні елементи серверних сценаріїв мовою PHP;
- правила взаємодії серверних сценаріїв мовою PHP з сервером БД MySQL;

уміти:

створити сайт (мінімум з 5 сторінок);
 використати засоби каскадних таблиць стилів для оформлення сторінки;
 використовувати основні об'єкти браузера та основні наперед визначені об'єкти JavaScript;
 написати нескладний сценарій обробки події;
 створити серверний сценарій мовою PHP та забезпечити його взаємодію з БД на сервері БД MySQL.

Бути здатними (компетенції):

алгоритмізувати та програмувати окремі підсистеми ІС на сучасних платформах розроблення програмних продуктів, орієнтованих на Internet-технології.

2. Тематичний план навчальної дисципліни

При вивченні дисципліни "Web-програмування" студент має ознайомитися з програмою дисципліни, її структурою, формами та методами навчання, видами та методами контролю знань.

Тематичний план дисципліни "Web-програмування" складається з двох модулів, кожний з яких об'єднує у собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками.

Навчальний процес здійснюється у таких формах: лекційні та лабораторні заняття, індивідуальна науково-дослідна робота, самостійна робота студента. Структура залікового кредиту дисципліни наведена у табл. 2.

Таблиця 2

Структура залікового кредиту навчальної дисципліни

Теми	Кількість годин			
	Лекції	Лабораторні заняття	Індивідуальна робота	Самостійна робота
Модуль 1 Основи Web-програмування на боці клієнта				
Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS	4	8		10
Тема 2. Розробка Web-сторінок за допомогою мови JavaScript	4	8		12
Разом годин за модулем	8	16	6	22
Модуль 2. Основи Web-програмування на боці сервера				
Тема 3. Основи мови PHP	4	4		8
Тема 4. Основи роботи з MySQL	2	4		6
Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL	2	8		8
Разом годин за модулем	8	16	10	22
Всього годин	16	32	16	44

3. Зміст навчальної дисципліни за модулями та темами

Модуль 1. Основи Web-програмування на боці клієнта

Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS

Розвиток та сучасний стан мови HTML. Структура документа HTML. Мінімальний документ. Теги для форматування документа. Посилання як основа гіпертексту. Списки. Використання графіки. Графічні формати. Таблиці в документі HTML. Фрейми в документі HTML. Переваги і недоліки фреймів. Обов'язкові вимоги при роботі з фреймами. Форми в документі HTML.

Каскадні таблиці стилів (CSS). Ідеологія і способи використання. Властивості форматування елементів засобами CSS: шрифт, колір і фон, форматування блоків. Властивості форматування елементів засобами CSS: візуальне форматування, форматування списків.

Тема 2. Розробка Web-сторінок за допомогою мови JavaScript

Об'єктна модель броузера. Поняття події. Процедури обробки події. Загальний огляд мови сценаріїв JavaScript. Синтаксис та особливості мови. Оператори JavaScript. Розгалуження і цикли. Оператори роботи з об'єктами у JavaScript. Особливості роботи з масивами. Функції у JavaScript. Створення об'єктів користувача у JavaScript. Організація передачі інформації між сторінками сайта засобами мови Javascript

Модуль 2. Основи Web-програмування на боці сервера

Тема 3. Основи мови PHP

Методи вбудовування PHP-коду. Вивід результатів роботи скрипта. Змінні у PHP. Типи даних та ініціалізація змінних. Перевірка існування змінних. Видалення змінних. Створення та використання констант. Оператори PHP: математичні, присвоювання, двійкові, конкатенації рядків, виклику зовнішніх програм. Перетворення типів даних. Масиви (одномірні, багатомірні, асоціативні) та робота з ними. Функції для роботи з рядками, числами, датою та часом. Функції користувача, глобальні та локальні змінні. Умовні оператори. Оператори циклів. Завершення виконання сценарію. Помилки у програмі. Змінні оточення. Глобальні масиви. Обробка даних форми.

Тема 4. Основи роботи з MySQL

Типи даних полів таблиць у MySQL. Створення бази даних у MySQL. Створення користувача бази даних. Створення таблиць. Додавання даних у таблицю. Оновлення записів у таблиці. Видалення записів з таблиці. Зміна властивостей таблиці. Вибірка записів. Видалення таблиці та бази даних.

Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL

Доступ до бази даних MySQL за допомогою PHP. Встановлення з'єднання із екземпляром сервера MySQL. Вибір бази даних. Виконання запиту до бази даних. Обробка результату запиту до бази даних.

4. Плани лекцій

Модуль 1. Основи Web-програмування на боці клієнта

Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS

1.1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML.

1.2. Розробка Web-сторінок за допомогою CSS.

Література: основна [7; 2; 4 – 9; 13], додаткова [20; 22].

Тема 2. Розробка Web-сторінок за допомогою мови JavaScript

2.1. Об'єктна модель браузера та загальний огляд мови сценаріїв JavaScript.

2.2. Типи даних, оператори, функції, класи та події у JavaScript.

Література: основна [4; 7], додаткова [15].

Модуль 2. Основи Web-програмування на боці сервера

Тема 3. Основи мови PHP

3.1. Введення у серверну мову сценаріїв PHP.

3.2. Масиви, функції, оператори PHP та обробка даних форм.

Література: основна [7; 10; 11], додаткова [19].

Тема 4. Основи роботи з MySQL.

4.1. Основи роботи з MySQL.

Література: основна [7; 10], додаткова [23].

Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL

5.1. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL.

Література: основна [1; 4; 7; 10], додаткова [19; 23].

5. Плани лабораторних занять

Лабораторне заняття – це організаційна форма навчального заняття, на якому студенти під керівництвом викладача формують уміння й навички з практичного застосування основних теоретичних положень навчальної дисципліни шляхом виконання завдань до лабораторних робіт.

Лабораторні заняття з дисципліни "Web-програмування" проводяться в спеціально обладнаному навчальному класі з використанням комп'ютерного устаткування пристосованого до начального процесу.

З метою підвищення якості навчального процесу, під час проведення лабораторного заняття призначається ще один викладач і навчальна група ділиться на дві підгрупи. Кожний студент працює самостійно, виконуючи індивідуальне завдання для лабораторного дослідження.

За результатами виконаної на занятті лабораторної роботи студенти оформлюють індивідуальні звіти з її виконання та захищають ці звіти перед викладачем. Результати виконання лабораторних досліджень оцінюються викладачем.

Тематика проведення лабораторних занять наведена у табл. 3.

План проведення лабораторних занять

Назва теми	Назва лабораторного заняття та питання що опрацьовуються	Кількість годин	Література
Модуль 1. Основи Web-програмування на боці клієнта			
Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS	Розробка Web-сторінок з використанням мови HTML. Створення персонального Web-сайту	4	Основна: [1; 2; 4 – 9; 13]. Додаткова: [20; 22]
	Розробка Web-сторінок з використанням HTML та CSS. Модифікація персонального Web-сайту	4	Основна: [1; 2; 4 – 9; 13]. Додаткова: [20; 22]
Тема 2. Розробка Web-сторінок за допомогою мови JavaScript	Розробка динамічних Web-сторінок за допомогою мови JavaScript	4	Основна: [4; 7]. Додаткова: [15]
	Розробка корпоративного Web-сайту підприємства за допомогою мови JavaScript	4	Основна: [4; 7]. Додаткова: [15]
Модуль 2. Основи Web-програмування на боці сервера			
Тема 3. Основи мови PHP	Розробка динамічних Web-сторінок за допомогою мови PHP	4	Основна: [7; 10; 11]. Додаткова: [19]
Тема 4. Основи роботи з MySQL	Проектування бази даних і забезпечення прав доступу до неї	4	Основна: [7; 10]. Додаткова: [23]
Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL	Розробка корпоративного Web-сайту за допомогою мови PHP та бази даних MySQL	8	Основна: [1; 4; 7; 10; 11]. Додаткова: [19; 23]
Разом годин		32	

6. Самостійна робота студентів

6.1. Основні форми самостійної роботи студентів

Для опанування матеріалу дисципліни "Web-програмування" окрім лекційних та лабораторних занять, тобто аудиторної роботи, значну увагу необхідно приділяти самостійній роботі.

Основні форми самостійної роботи студента:

1. Вивчення лекційного матеріалу.
2. Вивчення окремих тем або питань, що передбачені для самостійного опрацювання.
3. Вивчення основних термінів та понять за темами дисципліни.
4. Підготовка до лабораторних занять.
5. Контрольна перевірка кожним студентом знань за питаннями для самодіагностики.
6. Підготовка до проміжного та підсумкового модульного контролю.
7. Систематизація вивченого матеріалу перед іспитом.
8. Виконання індивідуального завдання.
9. Оформлення звітів з лабораторних робіт.
10. Робота з опрацювання та вивчення рекомендованої літератури.

Зразки завдань для позааудиторної самостійної роботи студента

Зразки завдань для самопідготовки під час позааудиторної самостійної роботи студента за трьома рівнями складності: базовому, творчому та інноваційному.

1. Базовий рівень складності:

1. Перетворити документ Word у веб-сторінку.
2. Перетворити презентацію у веб-сторінку.

2. Творчий рівень складності:

1. Створити веб-сторінку з використанням шаблонів Word.
2. Відіслати документ Word електронною поштою як додаток.

3. Інноваційний рівень складності:

1. Опублікувати сторінки на особистому веб-сайті.
2. Прорекламувати в Інтернеті власний веб-сайт.

Критерії оцінювання самостійної позааудиторної роботи

Завдання базового рівня оцінюються так:

"відмінно" – правильно виконане практичне завдання на ПК;

"добре" – правильно, але із незначними помилками виконане практичне завдання на ПК;

"задовільно" – не повністю виконане практичне завдання на ПК;

"незадовільно" – зовсім не виконане практичне завдання на ПК.

Завдання творчого рівня оцінюються так:

"відмінно" – правильно, із творчим підходом виконане практичне завдання на ПК;

"добре" – правильно, із творчим підходом, але із незначними помилками виконане практичне завдання на ПК;

"задовільно" – не повністю виконане практичне завдання на ПК;

"незадовільно" – зовсім не виконане практичне завдання на ПК.

Завдання інноваційного рівня оцінюються так:

"відмінно" – завдання виконане на високому інноваційному рівні; підібраний цікавий текстовий та графічний матеріал; були застосовані гіперпосилання та елементи управління, а також графіка та анімація;

"добре" – завдання виконане на високому інноваційному рівні; підібраний цікавий текстовий та графічний матеріал; були застосовані елементи графіки та анімації; не були застосовані гіперпосилання, елементи управління;

"задовільно" – завдання виконане, але підібраний нецікавий текстовий та графічний матеріал; не були застосовані гіперпосилання, елементи управління, анімація;

"незадовільно" – завдання не виконане на достатньому інноваційному рівні.

6.2. Питання для самостійного опрацювання

Модуль 1. Основи Web-програмування на боці клієнта

Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS

1. Історія розвитку гіпертексту.
2. Переваги HTML.
3. Універсальний ідентифікатор – URI.
4. Схеми адресації ресурсів в Internet.
5. Принципи роботи HTTP.
6. Особливості HTML 5.0. шари.
7. Використання шарів в HTML.
8. Підходи щодо оптимізації HTML-сторінок.
9. HTML і Cookies.
10. Основні відмінності різних DTD.
11. DTD та вплив на відображення сторінки.

Література: основна [1; 3; 4; 6; 7; 12], додаткова [1 – 4; 7].

Тема 2. Розробка Web-сторінок за допомогою мови JavaScript

1. Динамічні форми.
2. Читання та запис рядків у файли cookie.
3. Перетворення символів строки в Unicode та у зворотному напрямку.

4. Кодування та декодування URL.
5. Кодування та декодування строк у форматі Base64.
6. Обчислювання тригонометричних функцій.
7. Імітування хеш-таблиці.
8. Обробка помилок сценарію.
9. Перевірка доступності cookie.
10. Вирівнювання моделі IE та W3C DOM.
11. Прив'язування звуків до подій.
12. Запобігання попаданню сторінки до історії браузера.
13. Створення контекстного меню.
14. Меню на основі XML.

Література: основна [1; 4; 7], додаткова [15].

Модуль 2. Основи Web-програмування на боці сервера

Тема 3. Основи мови PHP

1. Відправка електронної пошти із сайту.
2. Автентифікація за допомогою PHP.
3. Заголовки HTTP та робота з ними.
4. Робота з файлами та каталогами.
5. Робота із графікою.
6. Способи перевірки відповідності формату файла, який завантажує користувач на сервер за допомогою форми (на боці клієнта) та PHP (на боці сервера).

Література: основна [7; 10; 11], додаткова [19].

Тема 4. Основи роботи з MySQL

1. Як організувати резервне копіювання БД MySQL?
2. У які формати здійснюється експортування БД MySQL?
3. Рішення щодо імпорту БД з неспівпаданням формату використаних полів дати з форматом дати, який встановлено на сервері MySQL.
4. Організація зв'язку таблиць у БД MySQL.
5. Параметри настроювання сервера MySQL.
6. Видалення таблиць за маскою.
7. Багатотабличні запити.
8. Організація повнотекстового пошуку в MySQL.

Література: основна [7], додаткова [23].

Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL

1. Як вирішуються проблеми з використанням різних кодових сторінок у MySQL та на Web-сторінках?

2. Як організувати розмежування прав доступу на сайті за допомогою MySQL та PHP?
 3. Оптимізація використання БД MySQL у PHP.
 4. Підключення до MySQL іншого домену.
 5. Організація пошуку в БД MySQL за допомогою PHP.
- Література:** основна [7; 10], додаткова [19; 23].

6.3. Контрольні запитання для самодіагностики

МОДУЛЬ 1. ОСНОВИ WEB-ПРОГРАМУВАННЯ НА БОЦІ КЛІЄНТА

Тема 1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS

1. З яких частин складається документ HTML?
2. У чому полягають правила сумісності синтаксису HTML з XML?
3. Чим відрізняється логічне і фізичне форматування тексту в документі HTML?
4. Поясніть різницю між правильним та дійсним документом.
5. Як використовується службова інформація в блоці заголовка документа HTML?
6. Перелічіть основні елементи структури сторінки.
7. Які типи списків можуть зустрічатися в документі HTML?
8. За яких умов для зображення краще обрати один з графічних форматів GIF або JPG?
9. Чому слід завжди визначати альтернативний текст для зображень?
10. Поясніть різницю між глобальним і локальним посиланням. Яка роль тега <base>?
11. Чому слід завжди передбачати частину <noframes> в документі, що описує фреймову структуру?
12. Які є переваги і недоліки використання фреймів у документі HTML?
13. Для чого можуть використовуватися форми в документі HTML? Наведіть приклади.
14. Для чого призначені каскадні таблиці стилів (CSS)?
15. Якими способами можна додати інформацію про стилі у документ HTML?
16. Як визначити, яке правило задання стилів діє в конкретному випадку?
17. Поясніть різницю між абсолютним і відносним позиціонуванням.
18. Які є рівні організації сайту?
19. З чого складається інтерфейс веб-сайта?

Література: основна [1; 2; 4 – 9; 13]; додаткова [20; 22].

Тема 2. Розробка Web-сторінок за допомогою мови JavaScript.

1. Які є способи включення сценаріїв JavaScript у документ HTML?
2. Чому сценарії не завжди слід описувати на початку документа?
3. Назвіть основні операції у мові сценаріїв JavaScript.
4. Порівняйте оператори розгалуження у мові сценаріїв JavaScript та у мові C++.
5. Порівняйте оператори циклів у мові сценаріїв JavaScript та у мові C++.
6. Назвіть оператори керування об'єктами у мові сценаріїв JavaScript.
7. Опишіть особливості масивів у мові сценаріїв JavaScript.
8. Як передаються параметри у функції у мові сценаріїв JavaScript?
9. Скільки складає час життя локальних і глобальних змінних сценарія JavaScript?
10. Назвіть наперед визначені об'єкти JavaScript.
11. Яким чином можна створити власний об'єкт у мові сценаріїв JavaScript?
12. Яким способом можна передати інформацію між сторінками сайта?

Література: основна [4; 7], додаткова [15].

Модуль 2. Основи Web-програмування на боці сервера

Тема 3. Основи мови PHP

1. Назвіть типи даних, які використовуються у PHP.
2. Яким чином вбудовуються PHP-скрипти у HTML?
3. Як оформлюються коментарі у PHP?
4. Як перевірити існування змінної?
5. Як видалити змінну?
6. Які різновиди запису умовного оператора існують в PHP?
7. Назвіть оператори циклу у PHP.
8. Чим відрізняється оператор вибору у PHP та у C++?
9. Як обробляти помилки у програмі?
10. Як об'явити функцію користувача?
11. Які глобальні масиви існують у PHP?

Література: основна [7; 10; 11], додаткова [19].

Тема 4. Основи роботи з MySQL.

1. Назвіть відомі Вам способи створення БД у MySQL.
2. Назвіть відомі Вам способи створення таблиць у MySQL.
3. Які типи даних існують у MySQL?
4. Чим відрізняється тип CHAR від типу VARCHAR?
5. Які формати типу дати існують і від чого залежать?
6. Чим відрізняється SQL, який використовується у MySQL, від стандартного SQL?
7. Які функції MySQL використовуються в SQL-запитах для організації повнотекстового пошуку в таблиці?
8. Які режими повнотекстового пошуку існують в MySQL?

9. Як створити індекс для повнотекстового пошуку?
10. Які особливості використання логічного режиму повнотекстового пошуку?

Література: основна [7; 10], додаткова [23].

Тема 5. Розробка Web-сторінок за допомогою мови PHP та СУБД MySQL

1. Як підключитися за допомогою PHP-скрипта до екземпляра сервера MySQL?
2. Яким чином обрати базу даних у підключеному екземплярі сервера MySQL?
3. Яким чином виконати SQL-запит в обраній базі даних MySQL?
4. У якому вигляді вертається результат запиту до бази даних MySQL?
5. Яким чином можна обробляти результат запиту до БД MySQL?
6. Як перевірити чи успішно було виконано запит з PHP до БД MySQL?
7. Чому дуже важливо виконувати перевірку успішності запиту?
8. Який метод передачі змінних форми до оброблювача більш надійний, особливо при авторизації користувачів?

Література: основна [1; 4; 7; 10], додаткова [19; 23].

6.4. Індивідуальне завдання

Індивідуальне завдання передбачає: систематизацію, закріплення, розширення теоретичних знань і практичних навичок із дисципліни та застосування їх при вирішенні конкретних виробничих ситуацій; розвиток навичок самостійної роботи з науково-технічною документацією.

Із дисципліни "WEB-програмування" видається студенту викладачем на початку вивчення дисципліни. Індивідуальне завдання виконується студентом самостійно. Студент має надати індивідуальне завдання для перевірки наприкінці семестру, але не пізніше терміну проведення підсумкового модульного контролю. Оцінка за виконання індивідуального завдання враховується при виставленні загальної оцінки з дисципліни.

Тематика індивідуального завдання має носити проблемний характер. Студент має право самостійно обрати тему та зміст роботи з обов'язковим її узгодженням з викладачем.

У протилежному випадку тема має бути запропонована викладачем (варіанти тем наведені нижче).

У процесі виконання індивідуального завдання студент має опрацювати не менш п'яти літературних джерел з посиланнями на використання певної інформації з них у тексті роботи. При цьому робота має носити творчий характер і бути спрямованою на вирішення певної проблеми чи на висловлення особистого погляду автора роботи на питання, яке розглядається в роботі.

Індивідуальне завдання складається з: титульної сторінки; змісту; вступу; основної частини; висновків; списку використаної літератури, додатків до індивідуального завдання (при необхідності).

Вступ має розкривати актуальність обраної студентом теми, її проблематику, мету написання роботи.

Основна частина роботи (може включати декілька підрозділів) має включати характеристику сучасного стану проблеми, опис нормативної бази, погляд різних авторів на цю проблему, позитивні та негативні наслідки проблеми.

Висновки мають включати обґрунтовані відомості студента щодо досягнення мети роботи, можливо, рекомендації щодо підвищення ефективності використання елементів та вузлів комп'ютерної схемотехніки.

Обсяг індивідуального завдання повинен становити у друкованому варіанті 10 – 15 сторінок. Орієнтовна кількість сторінок у розділах: вступ – 1с.; основна частина – 8 – 12 с.; висновки – 1 – 2 с.

Тематика індивідуального завдання

1. Аналіз поведінки браузерів при переключенні режимів за допомогою Dostupе.
2. Безпечне завантаження зображень на сервер по HTTP.
3. Особливості CMS Joomla.
4. Порівняльний аналіз PHP та ASP.
5. Ефективне використання .htaccess.
6. Використання різних PHP-фреймвок.
7. Digest-автентифікація в PHP.
8. Переваги та недоліки ООП у PHP.
9. Обґрунтування використання Smarty.
10. Тенденції розвитку сучасного Web-програмування.
11. Ефективність вибору засобів перевірки роботи із даними у формі між Javascript та PHP.
12. Фільтрування даних за допомогою PHP.
13. Організація безпечного пошуку на сайті.
14. Робота з електронною поштою за допомогою PHP.
15. Створення файлів за допомогою Javascript.
16. Побудова дерева ієрархій за допомогою PHP та MySQL.
17. PHP та регулярні вирази.
18. Посторінковий вивід у PHP.
19. Визначення поточної кількості користувачів, які присутні на сайті та ефективне використання цих даних.
20. Універсальні скрипти для обробки web-форм.
21. Фотомонтаж за допомогою PHP.
22. Організація повної статистики відвідувань сайта.

23. Робота з банерними системами.
24. Захист сайту від повного завантаження.
25. Захист web-форм від автоматичної обробки.
26. Створення гостьової книги за допомогою PHP та MySQL.
27. Захист адрес електронної пошти на web-сторінках.
28. Використання XML у PHP.
29. Сортування стовбців таблиці без перевантаження.
30. Стиснення даних за допомогою PHP.
31. Нові можливості PHP5-Zend Engine 2.0.
32. Захист сайту: блокування IP-адреси.
33. Аспектно-орієнтована web-розробка та PHP.
34. Можливості JpGraph.
35. Специфікація і функції DOM у PHP.
36. Хешування паролів у PHP.
37. Абстрактні класи та інтерфейси у PHP 5.
38. Реалізація шаблонів на PHP.
39. Створення автоматичних тестів та середовище phpUnit.
40. Створення таблиць Excel засобами PHP.
41. Використання XML Parser Functions для роботи із шаблонами.
42. Використання сокетів у PHP.
43. Особливості створення internet-магазину.
44. Система голосувань за допомогою PHP.
45. Особливості Javascript-бібліотеки Glassbox.
46. Підтвердження реєстрації у PHP.
47. RSS-парсер за допомогою PHP.
48. Аналіз та відображення логів.
49. Розширення PHP для роботи з mp3.
50. Створення водяних знаків за допомогою PHP.
51. Парсінг URL для "User Friendly URLs".
52. PHP-блокування.
53. Вразливість використання SSI (include) в PHP.
54. Робота зі строками у файлі за допомогою PHP.
55. Трансформування XML + XSLT за допомогою Sablotron.
56. Стиснення сторінок у PHP.
57. Реалізація тем на сайті за допомогою PHP.
58. Обчислення часу виконання SQL-запиту.
59. Використання сервісу Whois у PHP.
60. Обробка XML-документа за допомогою Javascript.

7. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, що винесені на поточний контроль тощо.

З теоретичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

1) індивідуальних консультацій, на яких студент отримує відповідь від викладача на конкретні запитання або пояснення певних теоретичних положень чи аспектів їх практичного застосування;

2) групових консультацій, на яких викладач розглядає типові приклади з використання концепцій об'єктно-орієнтованого програмування.

З практичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

1) індивідуальних консультацій, на яких викладач розглядає лабораторні завдання, стосовно яких виникли запитання у студента;

2) групових консультацій, на яких викладач розглядає практичні ситуації, які потребують колективного обговорення.

Індивідуально-консультативна робота для комплексної оцінки засвоєння програмного матеріалу проводиться у вигляді:

1) індивідуального захисту самостійних та індивідуальних завдань;

2) підготовки рефератів для виступу на науковому семінарі;

3) підготовки рефератів для виступу на науковій конференції.

8. Методики активізації процесу навчання

При викладанні навчальної дисципліни "Web-програмування" для активізації навчально-пізнавальної діяльності студентів передбачено застосування таких навчальних технологій, як: проблемні лекції, робота в малих групах, семінари-дискусії, презентації (табл. 4).

Таблиця 4

Використання навчальних технологій для активізації процесу навчання

Методики активізації процесу навчання	Практичне застосування навчальних технологій
1	2
Проблемні лекції направлено на розвиток логічного мислення студентів, коло питань теми обмежується двома-трьома ключовими моментами, використовується досвід закордонних навчальних закладів з роздачею студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. При читанні лекцій студентам даються питання для самостійного розміркування, на які лектор відповідає сам, не чекаючи відповідей студентів	Проблемна лекція з питання: "Напрямок розвитку Web-технологій" (за темою 1). Проблемна лекція з питань: "Вибір та використання серверних Web-технологій." (за темою 3)

1	2
Робота в малих групах дає змогу структурувати лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування	Робота в малих групах при розробленні корпоративного Web-сайта підприємства з використанням відповідних технологій (Л. р. №4 за модулем 1 та л. р. №7 за модулем 2)
Семінари-діскусії передбачають обмін думками і поглядами учасників з приводу даної теми, а також розвивають мислення, допомагають формувати погляди і переконання, виробляють вміння формулювати думки і висловлювати їх, вчать оцінювати пропозиції інших людей, критично підходити до власних поглядів	Проблемне повідомлення та дискусія при роботі студентів у проектній команді з питань: "Розроблення корпоративного Web-сайта підприємства з використанням всіх вивчених технологій". (Лабораторна робота №7 за модулем 2)
Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи	Презентація індивідуально вирішеного завдання

9. Система поточного та підсумкового контролю знань студентів

Система оцінювання знань, вмінь та навичок студентів передбачає виставлення оцінок за усіма формами проведення занять.

Перевірка та оцінювання знань студентів може проводитись в таких формах:

1. Оцінювання роботи студентів у процесі лабораторних занять.
2. Оцінювання виконання індивідуального завдання.
3. Проведення проміжного контролю.
4. Проведення модульного контролю.
5. Проведення підсумкового іспиту.

Загальна модульна оцінка складається з поточної оцінки, яку студент отримує під час лабораторних занять, оцінки за виконання індивідуального завдання та оцінки за виконання модульної контрольної роботи.

Загальна оцінка з дисципліни визначається як середнє арифметичне модульних оцінок та оцінки за результатами підсумкового іспиту.

Порядок поточного оцінювання знань студентів

Поточне оцінювання здійснюється під час проведення лабораторних занять і має на меті перевірку рівня підготовленості студента до виконання конкретної роботи. Об'єктами поточного контролю є:

- 1) активність та результативність роботи студента протягом семестру над вивченням програмного матеріалу дисципліни; відвідування занять;
- 2) виконання індивідуального завдання;
- 3) виконання проміжного контролю;
- 4) виконання модульного контрольного завдання.

Контроль систематичного виконання самостійної роботи та активності на лабораторних заняттях

Оцінювання проводиться за 12-бальною шкалою за такими критеріями:

1) розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються;

2) ступінь засвоєння матеріалу дисципліни;

3) ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;

4) уміння поєднувати теорію з практикою при розробці програм, розв'язанні задач, проведенні розрахунків при виконанні завдань, винесених для самостійного опрацювання, та завдань, винесених на розгляд в аудиторії;

5) логіка, структура, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" (10 –12 балів) ставиться за умови відповідності звіту з виконаного лабораторного завдання студента та його усної відповіді при захисті завдання усім п'ятьом зазначеним критеріям. Відсутність тієї або іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні лабораторних завдань увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу). Якщо якась із вимог не буде виконана, то оцінка на розсуд викладача буде знижена.

Проміжний модульний контроль

Проміжний модульний контроль рівня знань передбачає виявлення опанування студентом матеріалу лекційного модуля та вміння застосувати його для вирішення практичної ситуації і проводиться у вигляді тестування. При цьому тестове завдання може містити як запитання, що стосуються теоретичного матеріалу, так і запитання, спрямовані на вирішення невеличкого практичного завдання.

Тестове завдання містить запитання одиничного і множинного вибору різного рівня складності. Для оцінювання рівня відповідей студентів на тестові завдання використовуються критерій співвідношення кількості правильних відповідей та загальної кількості тестових запитань:

оцінка "відмінно" (12 балів) – 100 – 95% правильних відповідей;

оцінка "відмінно" (11 балів) – 94,99 – 90% правильних відповідей;

оцінка "відмінно" (10 балів) – 89,99 – 85% правильних відповідей;

оцінка "добре" (9 балів) – 84,99 – 80% правильних відповідей;

оцінка "добре" (8 балів) – 79,99 – 75% правильних відповідей;

оцінка "добре" (7 балів) – 74,99 – 70% правильних відповідей;

оцінка "задовільно" (6 балів) – 69,99 – 60% правильних відповідей;

оцінка "задовільно" (5 балів) – 59,99 – 50% правильних відповідей;

оцінка "задовільно" (4 бали) – 49,99 – 40% правильних відповідей;
оцінка "незадовільно" (3 бали) – 39,99 – 35% правильних відповідей;
оцінка "незадовільно" (2 – 1 бали) – менше ніж 35% правильних відповідей.

Тести для проміжного контролю обираються із загального переліку тестів за відповідними модулями.

Критерії оцінювання індивідуального завдання

Виконання індивідуального завдання є додатковою частиною самостійної роботи студента над навчальною дисципліною "Web-програмування". Мета індивідуального завдання – поглиблення теоретичних знань, які були набуті студентами в процесі вивчення дисципліни та придбання практичних вмінь та навичок з використання концепцій Web-програмування під час розроблення програм.

Індивідуальне завдання має бути виконано і подано на кафедру не пізніше зазначеної в навчальному плані дати.

Індивідуальне завдання оцінюється за критеріями:

самостійності виконання;

логічності та деталізації плану;

повноти й глибини розкриття теми;

наявності ілюстрацій (таблиці, рисунки, схеми тощо);

використання наукової інформації та відображення практичного досвіду;

якості оформлення.

Проведення поточно-модульного контролю

Поточно-модульний контроль здійснюється та оцінюється за двома складовими: практичний модульний контроль і лекційний (теоретичний) модульний контроль. Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час лабораторних занять та виконання індивідуального завдання. Лекційний модульний контроль здійснюється у формі тестів.

Для підведення підсумків роботи студентів із змістовного модуля виставляється підсумкова оцінка з поточно-модульного контролю, яка враховує оцінки за практичний модульний контроль і лекційний модульний контроль.

Таким чином, після вивчення тем 1 – 2 (модуль 1) студенти денної форми навчання виконують завдання до модуля 1; відповідно, після вивчення тем 3 – 5 (модуль 2) – завдання до модуля 2.

Приклад тестових завдань за модулем 1

Міністерство освіти і науки України
Харківський національний економічний університет
Кафедра інформаційних систем
Дисципліна: "Web-програмування"
Спеціальність "Інформаційні управляючі системи та технології"
Курс третій
Модуль №1

Завдання №1

1. Дайте відповідь на наступні запитання:

- а) Яким чином використовується SGML у HTML-кодi?
- б) Як використовується службова інформація в блоці заголовка документа HTML?
- в) Поясніть різницю між глобальним і локальним посиланням. Яка роль тега <base>?
- г) Які є способи включення сценаріїв JavaScript у документ HTML?

2. Оберіть вірний варіант твердження та обґрунтуйте свій вибір:

При реалізації сайт, який включає достатньо багато сторінок, найбільш ефективним способом використання CSS є:

- а) визначення стилю в елементі розмітки;
- б) розміщення опису стилю в заголовку документа в елементі STYLE;
- в) розміщення посилання на зовнішній опис через елемент LINK.

Затверджено на засіданні кафедри "Інформаційних систем"
протокол № від р.

Зав. кафедрою _____ проф. В. С. Пономаренко

Екзаменатор _____ доц. О. В. Щербаков

Приклад тестових завдань за модулем 2

Міністерство освіти і науки України
Харківський національний економічний університет
Кафедра інформаційних систем
Дисципліна: "Web-програмування"
Спеціальність "Інформаційні управляючі системи та технології"
Курс третій
Модуль №2

Завдання №1

1. Дайте відповідь на наступні запитання:

- а) Яким чином вбудовуються PHP-скрипти у HTML?
- б) Як перевірити існування змінної?
- в) Чим відрізняється тип CHAR від типу VARCHAR?
- г) Як підключитися за допомогою PHP-скрипта до екземпляра сервера MySQL?

2. Оберіть вірний варіант твердження та обґрунтуйте свій вибір:

8. Більш надійний метод передачі змінних форми до оброблювача, особливо при авторизації користувачів, є:

- а) GET;
- б) POST;
- в) PUT.

Затверджено на засіданні кафедри "Інформаційних систем"
протокол № _____ від _____ р.

Зав. кафедрою _____ проф. В. С. Пономаренко

Екзаменатор _____ доц. О. В. Щербаков

Проведення підсумкового іспиту

Іспит здійснюється у на комп'ютерах за екзаменаційними білетами. Екзаменаційний білет складається з двох завдань. Результати іспиту оцінюються за 12-бальною системою. Підсумкова оцінка за іспит є сумою оцінок за кожне завдання.

Приклад екзаменаційного білета з дисципліни

Міністерство освіти і науки України
Харківський національний економічний університет
Кафедра інформаційних систем
Дисципліна: "Web-програмування"
Спеціальність "Інформаційні управляючі
системи та технології"
Курс третій

Білет №1

Задача 1. Створити HTML-документ, який містить резюме студента. Обов'язково використати наступні можливості: форматування символів, заголовків. Вставити в документ таблицю такого виду:

За допомогою JavaScript вивести поточний день тижня.

Задача 2. За допомогою PHPMyAdmin розробити базу даних "Студенти", яка містить дані про прізвище, ім'я студента та його оцінку з екзамену. Розробити PHP-скрипт, який би виводив на екран прізвища відмінників.

Затверджено на засіданні кафедри "Інформаційних систем"
протокол № _____ від _____ р.

Зав. кафедрою _____ проф. В. С. Пономаренко

Екзаменатор _____ доц. О. В. Щербаков

Критерії оцінювання екзаменаційної роботи

Для оцінки рівня виконання студентами завдань використовуються такі критерії:

Кожне завдання оцінюється від 0 до 6 балів відповідно до такої шкали:

6 балів. Завдання виконано в повному обсязі. Програма працює правильно на усіх тестах. Інтерфейс та вихідний код програми задовольняють встановленим вимогам.

5 балів. Завдання виконано в повному обсязі. Програма працює правильно. Є невеликі зауваження до організації інтерфейсу користувача та (або) вихідного коду програми.

4 бали. Завдання в основному виконано. Програма працює правильно, але одна з її функціональних можливостей реалізована з порушенням вимог, вказаних у завданні.

3 бали. Завдання виконано, але не в повному обсязі. Програма працює, але не реалізована одна з функціональних вимог, вказаних у завданні, або дві та більше з функціональних можливостей програми реалізовані з порушенням вимог, вказаних у завданні.

2 бали. Завдання не виконано. Програма запускається, але не реалізовані дві та більше з функціональних вимог, вказаних у завданні.

1 бал. Програма не запускається, але є програмний код, розроблений студентом.

0 балів. Програма відсутня.

Підсумкова оцінка з дисципліни згідно з Методикою переведення показників успішності знань студентів Університету в систему оцінювання за шкалою ECTS конвертується в підсумкову оцінку за шкалою ECTS (табл. 5).

Таблиця 5

Переведення показників успішності знань студентів у систему оцінювання за шкалою ECTS

Відсоток студентів, які зазвичай успішно досягають відповідної оцінки	Оцінка за шкалою ECTS		Оцінка за бальною шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
10	відмінне виконання	A	12 – 10	відмінно
25	вище середнього рівня	B	10	
30	взагалі робота правильна, але з певною кількістю помилок	C	9 – 7	добре
25	непогано, але із значною кількістю помилок	D	6	задовільно
10	виконання задовольняє мінімальні критерії	D	5 – 4	
–	потрібне повторне перескладання	E	3	незадовільно
–	повторне вивчення дисципліни	FX	2 – 1	

10. Рекомендована література

10.1. Основна

1. Будилов В. А. Основы программирования для Интернета. – СПб.: БХВ-Петербург, 2003. – 736 с.
2. Глушков С. В. Программирование Web-страниц / С. В. Глушков, И. А. Жакин, Т. С. Хачиров. – Харьков: Фолио, 2005. – 390 с.
3. Калиновский А. И. Юзабилити: как сделать сайт удобным. – Минск: "Новое знание", 2005. – 220 с.
4. Лещев Д. Создание интерактивного web-сайта: учебный курс. – СПб.: Питер, 2003. – 544 с.
5. Ломов А. Ю. HTML, CSS, скрипты: практика создания сайтов. – СПб.: БХВ-Петербург, 2006. – 416 с.
6. Пономаренко В. С. Основы технологий Internet. Учебное пособие / В. С. Пономаренко, С. В. Минухин, И. А. Торохтий. – Харьков: Изд. ХГЭУ, 2001. – 256 с.
7. Прохоренок Н. HTML, Javascript, PHP и MySQL. Джентльменский набор Web-мастера. – СПб.: БХВ-Петербург, 2008. – 640 с.
8. Создание Web-страниц и Web-сайтов. Самоучитель / Под ред. В. Н. Печникова. – М.: Изд. Триумф, 2006. – 464 с.
9. Соколов С.А. HTML и CSS в примерах, типовых решениях и задачах. Профессиональная работа. – М.: ИД "Вильямс", 2007. – 416 с.
10. Томсон Л. Разработка Web-приложений на PHP и MySQL: Пер. с англ. – 2-е изд., испр. – СПб.: "ДиаСофтЮП", 2003. – 672 с.
11. Ульман Л. Основы программирования на PHP. – М.: ДМК Пресс, 2001. – 288 с.
12. Шмитт Кристофер CSS. Рецепты программирования. – СПб.: "БХВ-Петербург", 2007. – С. 592.
13. Эрик А. Мейер CSS-каскадные таблицы стилей: подробное руководство. – М.: "Символ", 2006. – С. 576.

10.2. Додаткова

14. Валентайн Ч. XHTML / Ч. Валентайн, К. Минник. – М.: ИД "Вильямс", 2001. – 480 с.
15. Дарнелл Р. JavaScript: справочник. – СПб.: "Питер", 2000. – 192 с.
16. Зельдман Д. Web-дизайн по стандартам. – СПб.: Изд. НТ Пресс, 2005. – 440 с.
17. Кирсанов Д. Веб-дизайн. – СПб.: "Символ-Плюс", 1999. – 376 с.
18. Круг С. Веб-дизайн. – СПб.: "Символ-Плюс", 2001. – 195 с.
19. Кузнецов М. Объектно-ориентированное программирование на PHP / М. Кузнецов, И. Симдянов. – СПб.: "БХВ-Петербург", 2007. – 608 с.
20. Моррисон М. HTML и XML. Быстро и эффективно. – СПб.: "Питер", 2005. – 302 с.
21. Нильсен Я. Веб-дизайн. – СПб.: "Символ-Плюс", 2000. – 512 с.

22. Хольцшлаг М. Э. Использование HTML и XHTML. – Спец. изд. – М.: ИД Вильямс. – 2003. – 728 с.

23. Шелдон Р. MySQL: базовый курс / Р. Шелдон, Д. Мойе. – М.: "Диалектика", 2007. – 880 с.

10.3. Ресурси мережі Internet

24. www.microsoft.com – сайт компанії Microsoft.

25. www.intuit.ru – Internet – інститут інформаційних технологій.

26. www.softtime.ru/bookphp/gl1_1.php – учебник по PHP 4.

27. www.mysql.ru/docs/man/ – справочное руководство по MySQL.

28. www.php.net – PHP: Hypertext Preprocessor.

29. php.ru – форум PHP программистов.

30. php.rus-phpnuke.com/ – учебник PHP.

31. html.manual.ru/ – HTML справочник.

32. www.spravkaweb.ru/ – справочник по Web-языкам.

Зміст

Вступ	1
1. Кваліфікаційні вимоги до студентів в галузі комп'ютерних наук	4
2. Тематичний план навчальної дисципліни	5
3. Зміст навчальної дисципліни за модулями та темами	6
Модуль 1 Основи Web-програмування на боці клієнта	6
Модуль 2. Основи Web-програмування на боці сервера	6
4. Плани лекцій	7
Модуль 1 Основи Web-програмування на боці клієнта	7
Модуль 2. Основи Web-програмування на боці сервера	7
5. Плани лабораторних занять	8
6. Самостійна робота студентів	10
6.1. Основні форми самостійної роботи студентів	10
6.2. Питання для самостійного опрацювання	11
6.3. Контрольні запитання для самодіагностики	13
6.4. Індивідуальне завдання	15
7. Індивідуально-консультативна робота	18
8. Методики активізації процесу навчання	18
9. Система поточного та підсумкового контролю знань студентів	19
10. Рекомендована література	26
10.1. Основна	26
10.2. Додаткова	26
10.3. Ресурси мережі Internet	27

НАВЧАЛЬНЕ ВИДАННЯ

Робоча програма
навчальної дисципліни
"WEB-ПРОГРАМУВАННЯ"
для студентів напряму підготовки "Комп'ютерні науки"
денної форми навчання

Укладачі: **Щербаков Олександр Всеволодович**
Огурцов Віталій Вячеславович

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Байдак В. В.**

Коректор **Мартовицька-Максимова В. А.**

План 2009 р. Поз. №231.

Підп. до друку Формат 60 × 90 1/16. Папір MultiCopy. Друк Riso.
Ум.-друк. арк. 1,75. Обл.-вид. арк. 2,19. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а
Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк №481 від 13.06.2001 р.

Робоча програма
навчальної дисципліни
"WEB-ПРОГРАМУВАННЯ"
для студентів напряму підготовки "Комп'ютерні науки"
денної форми навчання