

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 комп'ютерних наук та інформаційних систем

ЗАТВЕРДЖЕНО

вченою радою

(пост. п. 9 від «29» 09 20 22 р.)

Ректор



Аваторій МАЗАРАКІ

**ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНИХ  
ПРОДУКТІВ /  
SOFTWARE DEVELOPMENT TECHNOLOGIES**

**РОБОЧА ПРОГРАМА /  
COURSE OUTLINE**

освітній ступінь	бакалавр	/	bachelor
галузь знань	<u>12 Інформаційні технології</u>	/	<u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>

Київ 2022

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

Автори: Г. Т. САМОЙЛЕНКО, кандидат фізико-математичних наук, доцент,  
А.В. СЕЛІВАНОВА, старший викладач.

Робочу програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та інформаційних систем 19 липня 2022 р., протокол № 38

Рецензенти: Т.О. ФІЛІМОНОВА, канд.фіз.-матем. наук, доцент.  
С.П. КУДРЯВЦЕВА, канд.техн.н., провідний науковий співробітник Міжнародного науково-навчального центру інформаційних технологій та систем НАН України.

## **ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ / SOFTWARE DEVELOPMENT TECHNOLOGIES**

### **РОБОЧА ПРОГРАМА / COURSE OUTLINE**

освітній ступінь	бакалавр	/	Bachelor
галузь знань	<u>12 Інформаційні технології</u>	/	<u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/	<u>Information systems and technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/	<u>Information systems and technologies</u>

**1. СТРУКТУРА ДИСЦИПЛІНИ ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ  
(ТЕМАТИЧНИЙ ПЛАН)**

Назва теми	Кількість годин				Форми контролю
	Усього годин / кредитів	з них			
		лекції	лабораторні заняття	самостійна робота студентів	
1	2	3	4	5	6
Тема 1. Види програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 2. Життєвий цикл і етапи розробки програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 3. Стандартизація розробки програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 4. Сучасні методології розроблення програмних продуктів	15	2	2	11	О, ПСР, ПЛР
Тема 5. Архітектура програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 6. Розподілені архітектури програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 7. Стандарти розробки front-end	15	2	2	11	О, ПСР, ПЛР
Тема 8. Стандарти розробки backend	15	2	2	11	О, ПСР, ПЛР
Тема 9. Контейнеризація	15	2	2	11	О, ПСР, ПЛР
Тема 10. Система керування версіями	15	2	2	11	О, ПСР, МК, ПЛР
Тема 11. Якість програмного забезпечення	15	2	2	11	О, ПСР, ПЛР
Тема 12. Маркетинг програмних продуктів	15	2	2	11	О, ПСР, ПЛР
Разом	180/6	24	24	132	
Підсумковий контроль – екзамен					

**Умовні позначення: ПСР – перевірка самостійної роботи; МК – модульний контроль; ПЛР – перевірка лабораторної роботи; О – опитування.**

## 2. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ПРАКТИЧНИХ (СЕМІНАРСЬКИХ), ЛАБОРАТОРНИХ ЗАНЯТЬ, САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Результати навчання	Навчальна діяльність*	Робочий час студента, год
1	2	3
<p><b>Знати:</b> базові поняття технологій розробки програмного забезпечення</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про технології розробки програмного забезпечення в практичній діяльності.</p>	<p><b>Тема 1. Види програмного забезпечення.</b> <b>Лекція № 1. Види програмного забезпечення.</b> План лекції:</p> <ol style="list-style-type: none"> <li>1. Базові поняття. Види програмного забезпечення.</li> <li>2. Програмне забезпечення як виріб.</li> <li>3. Технологія розробки програмного забезпечення.</li> </ol> <p style="text-align: center;"><b>Список рекомендованих джерел</b> <i>Основний: 1[с.18-30]</i> <i>Додатковий: 1 [с.10-35],2[с.12-30]</i> <i>Інтернет-ресурси:1,2</i></p>	15
	<p style="text-align: center;"><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Види програмного забезпечення», підготовка до лабораторного заняття. Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Проблеми розробки складних програмних систем.</li> <li>2. Блочно-ієрархічний підхід до створення складних систем.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;"><b>Список рекомендованих джерел</b> <i>Основний: 1[с.18-30]</i> <i>Додатковий: 1 [с.10-35],2[с.12-30]</i> <i>Інтернет-ресурси:1,2</i></p>	11
	<p><b>Лабораторне заняття №1. Тема: «Види програмного забезпечення».</b> Завдання до заняття: проаналізувати конкретний сценарій та вибрати відповідне програмне забезпечення для його вирішення.</p>	2

	<ol style="list-style-type: none"> <li>1. Обрати сценарій, наприклад, автоматизація бухгалтерії, розробка веб-сайту, керування проектами тощо.</li> <li>2. Розглянути вимоги до програмного забезпечення для вибраного сценарію (функціональні та нефункціональні вимоги).</li> <li>3. Встановити критерії, за якими доцільно оцінювати програмне забезпечення (наприклад, ціна, функціональність, масштабованість, підтримка тощо).</li> <li>4. Провести дослідження ринку, знайти та порівняти різні програмні продукти, що відповідають вимогам та критеріям.</li> <li>5. Обрати програмне забезпечення, яке відповідає визначеним вимогам та критеріям.</li> <li>6. Підготувати короткий документ, в якому обґрунтовано вибір програмного забезпечення, враховуючи всі визначені аспекти.</li> <li>7. Підготувати звіт про виконання лабораторної роботи.</li> </ol>	
<p><b>Знати:</b> етапи розробки програмного забезпечення</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про етапи розробки програмного забезпечення в практичній діяльності.</p>	<p><b>Тема 2. Життєвий цикл і етапи розробки програмного забезпечення.</b></p> <p><b>Лекція № 2. Життєвий цикл і етапи розробки програмного забезпечення.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Процеси життєвого циклу програмного забезпечення.</li> <li>2. Процес розробки програмного забезпечення.</li> <li>3. Еволюція моделей життєвого циклу програмного забезпечення.</li> <li>4. Каскадна модель (waterflow model).</li> <li>5. Ітеративна модель (Iterative and incremental development) – модель з проміжним контролем.</li> <li>6. Спіральна модель.</li> </ol> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 1[с.35-50]</i></p> <p><i>Додатковий:1 [с.18-30]</i></p> <p><i>Інтернет-ресурси:1,2,3</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Життєвий цикл і етапи розробки програмного забезпечення», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі</p>	<p>11</p>

	<p>опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Етапи розробки програмного забезпечення при структурному підході до програмування: Стадія «Технічне завдання» Вибір моделі життєвого циклу.</li> <li>2. Визначення версій. Планування випуску версій. Розробка шаблону.</li> <li>3. Стадія «Ескізний проект».</li> <li>4. Стадія «Технічний проект».</li> <li>5. Стадія «Реалізація». Розробка документації.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;"><b>Список рекомендованих джерел</b></p> <p style="text-align: center;"><i>Основний: 1[с.35-50]</i></p> <p style="text-align: center;"><i>Додатковий:1 [с.18-30]</i></p> <p style="text-align: center;"><i>Інтернет-ресурси:1,2,3</i></p>	
	<p><b>Лабораторне заняття №2. Тема: «Життєвий цикл і етапи розробки програмного забезпечення».</b></p> <p><i>Завдання до заняття: дослідити етапи проектування програмного забезпечення</i></p> <ol style="list-style-type: none"> <li>1. Обрати конкретний проект для розробки програмного забезпечення (наприклад, мобільний додаток, веб-сайт, система управління тощо).</li> <li>2. Створити план розробки, включаючи визначення вимог, аналіз конкурентів, складання календарного графіка, визначення ресурсів тощо.</li> <li>3. Провести аналіз вимог, визначити функціональні та нефункціональні вимоги до програмного продукту.</li> <li>4. Створити дизайн архітектури програмного продукту, визначити алгоритми та структури даних.</li> <li>5. Підготувати звіт про виконання лабораторної роботи.</li> </ol>	2
<p style="text-align: center;"><b>Знати:</b> стандарти організації IEEE</p> <p style="text-align: center;"><b>Вміти:</b></p>	<p style="text-align: center;"><b>Тема 3. Стандартизація розробки програмного забезпечення.</b></p> <p style="text-align: center;"><b>Лекція № 3. Стандартизація розробки програмного забезпечення.</b></p>	15

<p>використовувати отримані теоретичні знання про міжнародні стандарти ISO в практичній діяльності.</p>	<p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Міжнародні стандарти ISO.</li> <li>2. Стандарти організації IEEE – Institute of Electrical and Electronics Engineers.</li> </ol> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 1[с.52-80]</i></p> <p><i>Додатковий: 2 [с.35-60], 3 [с.12-50]</i></p> <p><i>Інтернет-ресурси: 1, 2, 3</i></p>	<p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Стандартизація розробки програмного забезпечення», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Стандарт зрілості компанії-розробника програмного забезпечення CMM - Capability Maturity Model.</li> <li>2. Стандарт SPICE.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 1[с.52-80]</i></p> <p><i>Додатковий: 2 [с.35-60], 3 [с.12-50]</i></p> <p><i>Інтернет-ресурси: 1, 2, 3</i></p>	<p>11</p>
	<p><b>Лабораторне заняття №3. Тема:</b> «Стандартизація розробки програмного забезпечення».</p> <p><b>Завдання до заняття:</b> дослідити правила форматування коду, коментування та документації.</p> <ol style="list-style-type: none"> <li>1. Аналіз проекту: визначити основні особливості проекту, його розмір, фреймворки, які будуть використовуватися, та основні потреби команди.</li> <li>2. Стандарти кодування: дослідити правила форматування коду (відступи, розміщення дужок, розмір рядка тощо) згідно зі специфікаціями індустрії.</li> <li>3. Правила коментування: проаналізувати правила для коментування коду. Визначити,</li> </ol>	<p>2</p>

	<p>яка інформація має бути включена в коментарях до функцій, методів та складних частин коду.</p> <p>4. Документація коду: проаналізувати JSDoc та аналогічні інструменти для створення документації. Визначити, які блоки коментарів мають бути використані для опису класів, функцій, методів, їх параметрів та повернутих значень.</p> <p>5. Підготувати звіт про виконання лабораторної роботи.</p>	
<p><b>Знати:</b> основні поняття методологій розроблення програмних продуктів</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про жорсткі та гнучкі стратегії в методологіях програмування в практичній діяльності.</p>	<p><b>Тема 4. Сучасні методології розроблення програмних продуктів.</b></p> <p><b>Лекція № 4. Сучасні методології розроблення програмних продуктів.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. CASE-засоби та нотації моделювання програмних систем.</li> <li>2. Жорсткі та гнучкі стратегії в методологіях програмування. Методологія Rational Unified Process.</li> <li>3. Методологія Microsoft Solution Framework (MSF).</li> </ol> <p><b>Список рекомендованих джерел</b>  <i>Основний: 1 [с.52-80], 2 [с.60-92]</i>  <i>Додатковий: 3[с.22-60]</i>  <i>Інтернет-ресурси: 2,3</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Сучасні методології розроблення програмних продуктів», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Методологія eXtreme Programming (XP).</li> <li>2. Гнучке розроблення ПЗ на основі Agile.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b>  <i>Основний: 1 [с.52-80], 2 [с.60-92]</i></p>	<p>11</p>



	<p><i>Додатковий: 3[с.22-60]</i> <i>Інтернет-ресурси: 2,3</i></p> <p><b>Лабораторне заняття №4.</b> <i>Тема: «Сучасні методології розроблення програмних продуктів».</i></p> <p><i>Завдання до заняття: організувати та провести роботу з Agile методологією при створенні програмного продукту.</i></p> <ol style="list-style-type: none"> <li><i>1. Вибір проекту: обрати конкретний проект для розробки програмного продукту, який буде використовуватися для впровадження Agile.</i></li> <li><i>2. Тім-склад та ролі: сформувати команду для проекту, призначити ролі (Scrum Master, Product Owner, розробники тощо) та визначити їх обов'язки.</i></li> <li><i>3. Створення Backlog: розробити Product Backlog, перелік всіх завдань та функціоналу, який необхідно реалізувати в рамках проекту.</i></li> <li><i>4. Спринти та планування: поділити роботу на спринти, визначити завдання для першого спринту, зробити оцінку часу на їх виконання.</i></li> <li><i>5. Реалізація та збірка: провести реалізацію завдань першого спринту відповідно до угод та узгоджень.</i></li> <li><i>6. Ретроспектива: після закінчення першого спринту провести ретроспективу, обговорити, що пройшло добре, що можна покращити.</i></li> <li><i>7. Спланувати наступні спринти.</i></li> <li><i>8. Підготувати звіт про виконання лабораторної роботи.</i></li> </ol>	2
<p><b>Знати:</b> основні патерни проектування</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про патерни в практичній діяльності.</p>	<p><b>Тема 5. Архітектура програмного забезпечення.</b></p> <p><b>Лекція № 5. Архітектура програмного забезпечення.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Архітектура програмного забезпечення.</li> <li>2. Стандарти опису архітектури.</li> <li>3. Шаблони проектування.</li> <li>4. Патерни.</li> </ol> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 1[с.82-120]</i> <i>Додатковий: 2 [с.35-60],3 [с.12-50]</i> <i>Інтернет-ресурси:3</i></p>	15  2

	<p style="text-align: center;"><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Архітектура програмного забезпечення», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Паттерни створення: Singleton, Factory Method.</li> <li>2. Паттерни структури: Adapter, Decorator, Observer, Strategy.</li> <li>3. Паттерни архітектури: MVC (Model-View-Controller), Dependency Injection.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;"><b>Список рекомендованих джерел</b></p> <p style="text-align: center;"><i>Основний: 1[с.82-120]</i></p> <p style="text-align: center;"><i>Додатковий: 2 [с.35-60],3 [с.12-50]</i></p> <p style="text-align: center;"><i>Інтернет-ресурси:3</i></p>	11
	<p><b>Лабораторне заняття №5. Тема:</b> «Архітектура програмного забезпечення».</p> <p><b>Завдання до заняття:</b> розробити архітектуру системи для онлайн магазину, враховуючи функціональні та нефункціональні вимоги.</p> <ol style="list-style-type: none"> <li>1. <i>Аналіз вимог:</i> проаналізувати вимоги до системи: типи товарів, розмір бази клієнтів, способи оплати та доставки, адміністративна панель тощо.</li> <li>2. <i>Проектування архітектури:</i> визначити компоненти системи (клієнтський інтерфейс, сервер, база даних, підсистема адміністрування тощо). Обрати архітектурний стиль (наприклад, клієнт-сервер, мікросервісна архітектура). Створити діаграму компонентів та взаємодій між ними.</li> <li>3. <i>Вибір технологій:</i> обрати технології для реалізації кожного компонента системи (наприклад, React для фронтенду, Node.js для сервера, PostgreSQL для бази даних тощо).</li> <li>4. <i>Масштабованість та безпека:</i> розробити</li> </ol>	2

	<p><i>стратегію масштабування системи (горизонтальне/вертикальне масштабування). Визначити методи захисту даних та безпеки системи (автентифікація, шифрування, захист від атак тощо).</i></p> <p><i>5. Прототипування: розробити прототип однієї або декількох ключових частин системи для перевірки архітектурних рішень.</i></p> <p><i>6. Підготувати звіт про виконання лабораторної роботи.</i></p>	
<p><b>Знати:</b> особливості розподілених архітектур програмного забезпечення</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про розподілені архітектури в практичній діяльності.</p>	<p><b>Тема 6. Розподілені архітектури програмного забезпечення.</b></p> <p><b>Лекція № 6. Розподілені архітектури програмного забезпечення.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Клієнт-Серверна архітектура (Client-Server Architecture).</li> <li>2. Мікросервісна архітектура (Microservices Architecture).</li> <li>3. Хмарна архітектура (Cloud-Based Architecture).</li> </ol> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 1 [с.82-120],2 [с.80-112]</i></p> <p><i>Додатковий: 1 [с.60-92],2 [с. 35-60]</i></p> <p><i>Інтернет-ресурси: 3,4,5</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Розподілені архітектури програмного забезпечення», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Системи, засновані на повідомленнях (Message-Oriented Middleware).</li> <li>2. Peer-to-Peer (P2P) архітектура.</li> <li>3. Сервісно-орієнтована архітектура (Service-Oriented Architecture, SOA).</li> <li>4. Event-Driven Architecture (EDA).</li> <li>5. Space-Based Architecture (Space-Based Computing).</li> <li>6. Data-Centric Architecture.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на</p>	<p>11</p>



	<p>дотримання CSS-стандартів для підтримки рефакторингу.</p> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 2 [с.80-112]</i></p> <p><i>Додатковий: 2 [с.60-82], 3 [с. 35-60]</i></p> <p><i>Інтернет-ресурси: 3</i></p>	
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Стандарти розробки front-end», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Розробка з використанням компонентів (Component-Based Development).</li> <li>2. Responsive Web Design (RWD): розробка веб-інтерфейсу, що ефективно працює на різних пристроях та екранах.</li> <li>3. Accessibility (доступність).</li> <li>4. Performance Optimization (оптимізація продуктивності).</li> <li>5. Code Versioning (керування версіями коду).</li> <li>6. Використання систем керування версіями для ведення історії змін коду та спільної роботи в команді.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b></p> <p><i>Основний: 2 [с.80-112]</i></p> <p><i>Додатковий: 2 [с.60-82], 3 [с. 35-60]</i></p> <p><i>Інтернет-ресурси: 3</i></p>	11
	<p><b>Лабораторне заняття №7. Тема:</b> «Стандарти розробки front-end».</p> <p><i>Завдання до заняття: розробити користувацький інтерфейс для онлайн магазину електроніки, забезпечуючи зручність вибору товарів та процесу оформлення замовлення.</i></p> <p><i>1. Аналіз вимог: ознайомитись з вимогами до інтерфейсу: типи товарів, функціонал пошуку, сторінки товарів, корзина покупок,</i></p>	2

	<p><i>процес оформлення замовлення.</i></p> <p>2. <i>Створення макетів: розробити скетчі або wireframes сторінок: головна сторінка, сторінка категорій товарів, окрема сторінка товару, корзина покупок, форма замовлення. Обрати структуру та розташування елементів на сторінках.</i></p> <p>3. <i>Вибір технологій: обрати технології для розробки фронтенду (HTML, CSS, JavaScript). Розглянути використання бібліотек або фреймворків (наприклад, React, Vue.js).</i></p> <p>4. <i>Реалізація інтерфейсу: реалізувати користувацький інтерфейс на основі розроблених макетів. Забезпечити адаптивність інтерфейсу для різних пристроїв (мобільні та настільні).</i></p> <p>5. <i>Тестування: перевірити функціонал на різних браузерах та пристроях.</i></p> <p>6. <i>Підготувати звіт про виконання лабораторної роботи.</i></p>	
<p><b>Знати:</b> особливості стандартів розробки backend</p> <p><b>Вміти:</b> використовувати отримані знання про стандарти розробки в практичній діяльності.</p>	<p><b>Тема 8. Стандарти розробки backend</b> <b>Лекція № 8. Стандарти розробки backend</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. RESTful API Design: використання REST або подібного до REST підходу для створення API.</li> <li>2. Захист даних та забезпечення аутентифікації та авторизації.</li> <li>3. Використання протоколів HTTPS для шифрування з'єднань.</li> <li>4. Використання Систем Керування Базами Даних (Database Management Systems - DBMS).</li> </ol> <p><b>Список рекомендованих джерел</b> Основний: 2[с.120-142] Додатковий: 2 [с.60-82],3 [с. 35-60] Інтернет-ресурси: 3</p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Стандарти розробки backend», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет.</p>	<p>11</p>

	<p>Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Логування (Logging): Логування подій та помилок для полегшення відладки та моніторингу.</li> <li>2. Документація коду (Code Documentation).</li> <li>3. Масштабованість (Scalability).</li> <li>4. Архітектурні патерни (Architectural Patterns).</li> <li>5. Керування Залежностями (Dependency Management).</li> <li>6. Стандартизація форматів обміну даними.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;"><b>Список рекомендованих джерел</b></p> <p style="text-align: center;"><i>Основний: 2[с.120-142]</i></p> <p style="text-align: center;"><i>Додатковий: 2 [с.60-82],3 [с. 35-60]</i></p> <p style="text-align: center;"><i>Інтернет-ресурси: 3</i></p>	
	<p><b>Лабораторне заняття №8. Тема: «Стандарти розробки backend».</b></p> <p><i>Завдання до заняття: розробити backend частину для онлайн-магазину електроніки, забезпечуючи доступ до даних про товари та здійснення замовлень.</i></p> <ol style="list-style-type: none"> <li>1. <i>Аналіз вимог: ознайомитись з вимогами до API: отримання списку товарів, детальна інформація про товар, створення замовлення, обробка платежів тощо.</i></li> <li>2. <i>Вибір технологій та структури: визначити технології для розробки backend (наприклад, Node.js, Python, Java). Структурувати проекти API (розділення за модулями, роутинг).</i></li> <li>3. <i>Реалізація API: створити ендпоінти для отримання списку товарів, детальної інформації про товар, створення та обробки замовлень. Реалізувати можливість аутентифікації та авторизації користувачів (якщо це потрібно).</i></li> <li>4. <i>Тестування: використати тести для перевірки працездатності та правильності роботи ендпоінтів.</i></li> <li>5. <i>Документація та безпека: розробити документацію API для розробників.</i></li> <li>6. <i>Підготувати звіт про виконання</i></li> </ol>	2

	<i>лабораторної роботи.</i>	
<p><b>Знати:</b> поняття контейнеризації</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про Docker Compose в практичній діяльності.</p>	<p><b>Тема 9. Контейнеризація.</b> <b>Лекція № 9. Контейнеризація.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Контейнеризація.</li> <li>2. Віртуальні машини і контейнери.</li> <li>3. Платформа Docker та її компоненти.</li> <li>4. Образ і контейнер.</li> <li>5. Реєстр образів. Базові команди Docker.</li> </ol> <p><b>Список рекомендованих джерел</b> <i>Основний: 2[с.120-142]</i> <i>Додатковий: 3 [с. 65-80]</i> <i>Інтернет-ресурси: 4,5</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Контейнеризація», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Dockerfile. Побудова Docker-образу.</li> <li>2. Інструмент Docker Compose.</li> <li>3. Використання оркестратора Docker Compose для запуску багатоконтейнерних додатків.</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b> <i>Основний: 2[с.120-142]</i> <i>Додатковий: 3 [с. 65-80]</i> <i>Інтернет-ресурси: 4,5</i></p>	11
	<p><b>Лабораторне заняття №9. Тема:</b> <b>«Контейнеризація»</b></p> <p><i>Завдання до заняття: контейнеризувати веб-додаток за допомогою Docker.</i></p> <ol style="list-style-type: none"> <li>1. Розробка веб-додатку: розробити простий веб-додаток (наприклад, сайт на Node.js або Flask), який можна буде контейнеризувати.</li> <li>2. Створення Dockerfile: створити Dockerfile для розробленого веб-додатку, налаштувати середовище контейнера.</li> <li>3. Розробка контейнера: розробити контейнер</li> </ol>	2



	<p>за допомогою <i>Dockerfile</i>.</p> <p>4. Тестування контейнера: запустити контейнер та перевірити роботу вашого веб-додатку в ізольованому середовищі. Перевірити, що веб-додаток працює так само, як і без контейнеризації.</p> <p>5. Підготувати звіт про виконання лабораторної роботи.</p>	
<p><b>Знати:</b> особливості системи керування версіями</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про керування версіями в практичній діяльності.</p>	<p><b>Тема 10. Система керування версіями.</b> <b>Лекція № 10. Система керування версіями.</b> План лекції:</p> <ol style="list-style-type: none"> <li>1. Базові поняття. Створення репозиторію.</li> <li>2. Операції Git при внесенні змін.</li> <li>3. Робота з гілками. Запити на злиття гілки. Видалення гілки. Коміти злиття.</li> <li>4. Конфлікти злиття та їх вирішення.</li> <li>5. Виключення відстеження файлів.</li> </ol> <p><b>Список рекомендованих джерел</b> <i>Основний: 2 [с.145-170]</i> <i>Додатковий: 3 [с. 65-80]</i> <i>Інтернет-ресурси: 6</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Система керування версіями», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Збереження незавершених змін (git stash).</li> <li>2. Навігація по історії.</li> <li>3. Скасування або внесення змін до вже зроблених комітів.</li> <li>4. Злиття гілок (у власну гілку).</li> </ol> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b> <i>Основний: 2 [с.145-170]</i> <i>Додатковий: 3 [с. 65-80]</i> <i>Інтернет-ресурси: 6</i></p>	<p>11</p>

	<p><b>Лабораторне заняття №10. Тема: «Система керування версіями».</b>  <b>Завдання до заняття:</b></p> <ol style="list-style-type: none"> <li>1. Створити новий репозиторій на сервісі хостингу (наприклад, GitHub або GitLab).</li> <li>2. Клонувати репозиторій на локальний комп'ютер.</li> <li>3. Створити дві гілки - одну для розробки нового функціоналу (наприклад, feature/awesome-feature) та іншу для виправлення помилок (наприклад, bugfix/fix-bug).</li> <li>4. Внести зміни у кожну з гілок - додати файли, внести зміни в існуючі файли.</li> <li>5. Змінити один і той самий рядок або регіон коду в обох гілках, щоб викликати конфлікт.</li> <li>6. Об'єднати гілки, вирішивши конфлікти, застосовуючи зміни з обох гілок.</li> <li>7. Відправити зміни до віддаленого репозиторію та створити пулл-реквест для об'єднання робочої гілки з головною гілкою.</li> <li>8. Підготувати звіт про виконання лабораторної роботи.</li> </ol>	2
<p><b>Знати:</b>  особливості тестування програмного забезпечення</p> <p><b>Вміти:</b>  використовувати отримані теоретичні знання про тестування програмного забезпечення в практичній діяльності.</p>	<p><b>Тема11. Якість програмного забезпечення</b>  <b>Лекція № 11. Якість програмного забезпечення</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Забезпечення якості програмного забезпечення.</li> <li>2. Тестування програмного забезпечення. Методи оцінки якості тестування.</li> <li>3. Покриття коду: вимірювання ступеня покриття коду тестами (якість тестування).</li> <li>4. Метрики дефектів: аналіз кількості і серйозності виявлених дефектів.</li> </ol> <p><b>Список рекомендованих джерел:</b>  Основний: 1[с. 120-170]  Додатковий: 4 [с. 65-80]  Інтернет-ресурси:1</p>	15  2
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Якість програмного забезпечення», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі</p>	11

опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:

1. Вимірювання часу, який витрачається на підготовку, виконання тестів та аналіз результатів.
2. Ефективність тестування: оцінка ефективності тестових сценаріїв у виявленні дефектів. Стійкість до помилок.
3. Узгодженість тестів. Види тестів.
4. Модульні тести (Unit tests). Інтеграційні тести (Integration tests). Функціональні тести (Functional tests).
5. Тести відмови (Regression tests). Тести навантаження (Load tests). Тести безпеки (Security tests).
6. Тести на користувальницький інтерфейс (UI tests).
7. Верифікація програм.

Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.

#### **Список рекомендованих джерел**

*Основний: 1 [с. 120-170]*

*Додатковий: 4 [с. 65-80]*

*Інтернет-ресурси: 1*

#### ***Лабораторне заняття №11. Тема «Якість програмного забезпечення»***

*Завдання до заняття: розробити тестовий сценарій та провести тестування функціоналу веб-додатку для онлайн-магазину з метою забезпечення його надійності та коректності роботи.*

1. *Аналіз функціоналу: ретельно ознайомитись з функціоналом онлайн-магазину. Визначити основні функції, які потрібно протестувати (наприклад, додавання товарів у кошик, оформлення замовлення, оплата тощо).*
2. *Розробка тестового сценарію: створити тестовий сценарій для однієї з ключових функцій (наприклад, оформлення замовлення). Уточнити вхідні дані, очікувані результати та критерії успішності тесту.*
3. *Виконання тестів: виконати розроблений тестовий сценарій на реальному екземплярі*

	<p>веб-додатку. Зафіксувати результати виконання тесту, виявлені помилки та неполадки.</p> <p>4. Розширення тестового покриття: розробити додаткові тестові сценарії для інших важливих функцій магазину (наприклад, пошук товарів, фільтрація результатів тощо).</p> <p>5. Підготувати звіт про виконання лабораторної роботи.</p>	
<p><b>Знати:</b> особливості маркетингу програмних продуктів</p> <p><b>Вміти:</b> використовувати отримані теоретичні знання про маркетинг програмних продуктів в практичній діяльності.</p>	<p><b>Тема 12. Маркетинг програмних продуктів.</b> <b>Лекція № 12. Маркетинг програмних продуктів.</b></p> <p>План лекції:</p> <ol style="list-style-type: none"> <li>1. Дослідження ринку та цільової аудиторії, визначення їхніх потреб, вимог і конкурентного середовища.</li> <li>2. Створення стратегії продукту. Визначення унікальних пропозицій продукту (Unique Selling Proposition - USP) та ключових переваг, які роблять продукт привабливим для споживачів.</li> <li>3. Розробка стратегії позиціонування, яка визначає місце продукту на ринку порівняно з конкурентами.</li> </ol> <p><b>Список рекомендованих джерел:</b>  <i>Основний: 2 [с.145-170]</i>  <i>Додатковий: 4 [с. 110-180]</i>  <i>Інтернет-ресурси: 2</i></p>	<p>15</p> <p>2</p>
	<p><b>Самостійна робота</b></p> <p>Вивчення матеріалу до теми «Маркетинг програмних продуктів», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> <li>1. Канали просування: Вибір оптимальних каналів для реклами.</li> <li>2. Створення контенту. Залучення користувачів та отримання відгуків.</li> <li>3. Взаємодія з користувачами, залучення до співпраці та отримання відгуків для покращення продукту та його просування на ринку.</li> <li>4. Аналіз та оптимізація стратегій.</li> </ol>	<p>11</p>

	<p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p><b>Список рекомендованих джерел</b>  <i>Основний: 2 [с.145-170]</i>  <i>Додатковий: 4 [с. 110-180]</i>  <i>Інтернет-ресурси: 2</i></p>	
	<p><b>Лабораторне заняття №12.</b> <i>Тема: «Маркетинг програмних продуктів»</i></p> <p><i>Завдання до заняття: розробити стратегію маркетингу для нового програмного продукту з метою просування його на ринок та залучення цільової аудиторії.</i></p> <ol style="list-style-type: none"> <li><i>1. Аналіз ринку та цільової аудиторії: провести аналіз ринку програмного забезпечення відповідного сектору. Визначити цільову аудиторію для продукту, потреби та очікування.</i></li> <li><i>2. Позиціонування продукту: визначити унікальні переваги програмного продукту.</i></li> <li><i>3. Створення маркетингового плану: розробити стратегію просування продукту, визначити рекламні і маркетингові інструменти, які будуть використані для досягнення цілей.</i></li> <li><i>4. Створення контенту та ресурсів: розробити план створення контенту, який буде привертати увагу цільової аудиторії (статті, відео, інфографіка тощо).</i></li> <li><i>5. Визначити ресурси, необхідні для реалізації цього плану.</i></li> <li><i>6. Розробити метрики успіху та план їх вимірювання.</i></li> <li><i>7. Підготувати звіт про виконання лабораторної роботи.</i></li> </ol>	2
	<b>Разом</b>	<b>180</b>

\* +20% інтерактиву – зазначені курсивом

### 3.СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

#### Основний:

1. Шаховська Н. Б. Проектування інформаційних систем : навчальний посібник / Н. Б. Шаховська, В. В. Литвин ; за наук. ред. В. В. Пасічника ; М-во освіти і науки України. - Л. : Магнолія 2021. - 380 с.
2. Методи та новітні підходи до проектування, управління і застосування високопродуктивних ІТ-інфраструктур: монографія / Ю. В. Бойко, В. М. Волохов,

М. М. Глибовець, С. В. Єршов, С. Л. Кривий, С. Д. Погорілий, О. І. Ролік, С. Ф. Теленик, М. В. Ясочка // за ред. проф. А. В. Анісімова. – К.: ВПЦ «Київський університет», 2016. – 447 с.

#### **Додатковий:**

1. Катренко А.В., Системний аналіз об'єктів та процесів комп'ютеризації : навчальний посібник /А.В. Катренко Системний аналіз об'єктів та процесів комп'ютеризації Львів:"Новий світ-2000".-2003.-424с.
2. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
3. Гломозда Д. К., Проектування, системний аналіз і розробка корпоративних інформаційних систем : навчальний посібник / Гломозда Дмитро ; Нац. ун-т "Києво-Могилян. акад.". - Київ : [НаУКМА], 2015. - 95 с
4. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с

#### **Інтернет-ресурси:**

1. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Електронні текстові дані (1 файл: 2,02 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 192с.: [https://ela.kpi.ua/bitstream/123456789/33651/1/PIS\\_KL.pdf](https://ela.kpi.ua/bitstream/123456789/33651/1/PIS_KL.pdf)
2. Основи управління ІТ проектами [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки»/ КПІ ім. Ігоря Сікорського ; уклад.: В. О. Кузьмич, Р. А. Тараненко. – Електронні текстові дані (1 файл:1,998 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 75 с. [https://ela.kpi.ua/bitstream/123456789/34480/1/2019\\_Osnovy\\_upravlinnia.pdf](https://ela.kpi.ua/bitstream/123456789/34480/1/2019_Osnovy_upravlinnia.pdf)
3. <https://refactoring.guru/uk/design-patterns/catalog>
4. <https://kubernetes.io/>
5. <https://www.docker.com/>
6. <https://git-scm.com/>

\*- Курсивом виділені джерела, що є в бібліотеці ДТЕУ