



**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ  
УНІВЕРСИТЕТ**  
**Факультет інформаційних технологій**  
**Кафедра комп'ютерних наук та інформаційних систем**

**СИЛАБУС (SYLLABUS)**

**Дисципліна «Технології створення програмних продуктів /Software  
development technologies»**

**ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА**

Викладач	Самойленко Ганна Тимофіївна
Науковий ступінь	Кандидат фізико-математичних наук
Вчене звання	Доцент
Посада	Доцент кафедри комп'ютерних наук та інформаційних систем
Адреса кафедри	м.Київ, вул. Кіото 19, каб. Б-507, Б-526
E-mail	compdep@knute.edu.ua
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

**ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ**

<https://knute.edu.ua/file/MzEyMQ==/c12a9f74e87d9154696ca0f761da2e5c.pdf>

**Дотримання академічної доброчесності передбачає:**

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей);
- посилання на джерела інформації у разі використання не авторських ідей, розробок, тверджень, відомостей і т.п.;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної наукової діяльності, використанні методики досліджень і джерела інформації.

**Порушенням академічної доброчесності вважається:**

- академічний плагіат – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства;
- самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;
- фабрикація – вигадкування даних чи фактів, що використовуються в наукових дослідженнях;
- фальсифікація – свідомо зміна чи модифікація вже наявних даних, що стосуються наукових досліджень.

**За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до академічної відповідальності:**

- повторне проходження оцінювання (модульний контроль, іспит, залік тощо);
- повторне проходження відповідного освітнього компонента освітньо-професійної програми;
- відрахування з Університету;
- позбавлення наданих університетом пільг;
- відмова у присудженні відповідного ступеня вищої освіти;

## ПОЛІТИКА ЩОДО ВІДВІДУВАННЯ ЗАНЯТЬ

- відвідування занять є обов'язковим;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування та ін.) навчання може відбуватись в он-лайн формі за погодженням із викладачем дисципліни.

### ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<b>Назва дисципліни / тип дисципліни</b>	Технології створення програмних продуктів /за вибором
<b>Навчальний рік</b>	2023-2024
<b>Факультет</b>	Факультет інформаційних технологій
<b>Курс</b>	4
<b>Семестр</b>	7-8
<b>Освітній ступінь</b>	Бакалавр
<b>Галузь знань</b>	12 «Інформаційні технології»
<b>Спеціальність</b>	126 «Інформаційні системи та технології»
<b>Загальна характеристика</b>	Кількість годин –180 Кількість кредитів – 6 <b>Види занять:</b> лекції, лабораторні, самостійна робота. <b>Співвідношення аудиторних годин і годин самостійної роботи -</b> 48/132 <b>Мова викладання</b> – українська <b>Форма викладання</b> – очна
<b>Підсумковий контроль</b>	Екзамен
<b>Програмне забезпечення</b>	Visual Studio, Figma, Docker, Git
<b>Обладнання</b>	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
<b>Необхідні попередні дисципліни</b>	«Алгоритмізація та програмування»
<b>Методика вивчення</b>	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного і практично-прикладного характеру під час лекцій, лабораторних занять, самостійної роботи та вивчення першоджерел і навчально-методичної літератури.
<b>Мета і завдання</b>	<b>Метою</b> вивчення дисципліни «Технології створення програмних продуктів» є набуття теоретичних знань і практичних навичок з сучасних технологій створення програмних продуктів. <b>Завданням</b> вивчення дисципліни «Технології створення програмних продуктів» є вивчення моделей життєвого циклу програмного забезпечення, основних методологій та інструментів розробки програмного забезпечення, а також документування програмних продуктів.
<b>Місце дисципліни в освітньо-професійній програмі</b>	
<b>Загальні компетентності</b>	КЗ 3 Здатність до розуміння предметної області та професійної діяльності.
<b>Фахові компетентності (результати навчання)</b>	КС 1 Здатність аналізувати об'єкт проектування або функціонування та його предметну область. КС 3 Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної

	<p>структури, управління ними.</p> <p>КС 4 Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).</p>
<b>Програмні результати навчання</b>	<p>ПР 2 <b>Застосовувати</b> знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.</p> <p>ПР 3 <b>Використовувати</b> базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.</p> <p>ПР 5 <b>Аргументувати</b> вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; <b>мати навички</b> налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.</p> <p>ПР 6 <b>Демонструвати</b> знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.</p> <p>ПР 7 <b>Обґрунтовувати</b> вибір технічної структури та <b>розробляти</b> відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.</p> <p>ПР 9 <b>Здійснювати</b> системний аналіз архітектури підприємства та його ІТ інфраструктури, проводити розроблення та вдосконалення її елементної бази і структури.</p>

## ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Тема 1. Види програмного забезпечення.

Базові поняття. Види програмного забезпечення. Програмне забезпечення як виріб. Технологія розробки програмного забезпечення. Проблеми розробки складних програмних систем. Блочно-ієрархічний підхід до створення складних систем.

### Тема 2. Життєвий цикл і етапи розробки програмного забезпечення.

Процеси життєвого циклу програмного забезпечення. Процес розробки програмного забезпечення. Еволюція моделей життєвого циклу програмного забезпечення. Каскадна модель (waterflow model). Ітеративна модель (Iterative and incremental development) – модель з проміжним контролем. Спіральна модель. Етапи розробки програмного забезпечення при структурному підході до програмування: Стадія «Технічне завдання» Вибір моделі життєвого циклу. Визначення версій. Планування випуску версій. Розробка шаблону. Стадія «Ескізний проект». Стадія «Технічний проект». Стадія «Реалізація». Розробка документації.

### Тема 3. Стандартизація розробки програмного забезпечення.

Міжнародні стандарти ISO. Стандарти організації IEEE – Institute of Electrical and Electronics Engineers. Стандарт зрілості компанії-розробника програмного забезпечення CMM - Capability Maturity Model. Стандарт SPICE.

#### **Тема 4. Сучасні методології розроблення програмних продуктів.**

CASE-засоби та нотації моделювання програмних систем. Жорсткі та гнучкі стратегії в методологіях програмування. Методологія Rational Unified Process. Методологія Microsoft Solution Framework (MSF). Методологія eXtreme Programming (XP). Гнучке розроблення ПЗ на основі Agile.

#### **Тема 5. Архітектура програмного забезпечення.**

Архітектура програмного забезпечення. Стандарти опису архітектури. Шаблони проектування. Патерни. Паттерни створення: Singleton, Factory Method. Паттерни структури: Adapter, Decorator, Observer, Strategy. Паттерни архітектури: MVC (Model-View-Controller), Dependency Injection.

#### **Тема 6. Розподілені архітектури програмного забезпечення.**

Клієнт-Серверна архітектура (Client-Server Architecture). Системи, засновані на повідомленнях (Message-Oriented Middleware). Peer-to-Peer (P2P) архітектура. Сервісно-орієнтована архітектура (Service-Oriented Architecture, SOA). Event-Driven Architecture (EDA). Space-Based Architecture (Space-Based Computing). Data-Centric Architecture. Мікросервісна архітектура (Microservices Architecture). Хмарна архітектура (Cloud-Based Architecture).

#### **Тема 7. Стандарти розробки front-end.**

ECMAScript та JavaScript-стандарти. Використання стандарту ECMAScript та дотримання сучасних практик розробки JavaScript. Дотримання семантичних правил HTML, використання HTML5-елементів та атрибутів для підтримки функціоналу та структуризації документів. Використання CSS-препроцесорів та дотримання CSS-стандартів для підтримки рефакторингу. Розробка з використанням компонентів (Component-Based Development). Responsive Web Design (RWD): розробка веб-інтерфейсу, що ефективно працює на різних пристроях та екранах. Accessibility (доступність). Performance Optimization (оптимізація продуктивності). Code Versioning (керування версіями коду). Використання систем керування версіями для ведення історії змін коду та спільної роботи в команді.

#### **Тема 8. Стандарти розробки backend**

RESTful API Design: використання REST або подібного до REST підходу для створення API. Захист даних та забезпечення аутентифікації та авторизації. Використання протоколів HTTPS для шифрування з'єднань. Використання Систем Керування Базами Даних (Database Management Systems - DBMS). Логування (Logging): Логування подій та помилок для полегшення відладки та моніторингу. Документація коду (Code Documentation). Масштабованість (Scalability). Архітектурні патерни (Architectural Patterns). Керування Залежностями (Dependency Management). Стандартизація форматів обміну даними.

#### **Тема 9. Контейнеризація.**

Контейнеризація. Віртуальні машини і контейнери. Платформа Docker та її компоненти. Образ і контейнер. Реєстр образів. Базові команди Docker. Dockerfile. Побудова Docker-образу. Інструмент Docker Compose. Використання оркестратора Docker Compose для запуску багатоконтейнерних додатків.

#### **Тема 10. Система керування версіями.**

Базові поняття. Створення репозиторію. Операції Git при внесенні змін. Робота з гілками. Запити на злиття гілки. Видалення гілки. Коміти злиття. Конфлікти злиття та їх вирішення. Виключення відстеження файлів. Збереження незавершених змін (git stash). Навігація по історії. Скасування або внесення змін до вже зроблених комітів. Злиття гілок (у власну гілку).

### Тема 11. Якість програмного забезпечення

Забезпечення якості програмного забезпечення. Тестування програмного забезпечення. Методи оцінки якості тестування. Покриття коду: вимірювання ступеня покриття коду тестами (якість тестування). Метрики дефектів: аналіз кількості і серйозності виявлених дефектів. Вимірювання часу, який витрачається на підготовку, виконання тестів та аналіз результатів. Ефективність тестування: оцінка ефективності тестових сценаріїв у виявленні дефектів. Стійкість до помилок. Узгодженість тестів. Види тестів. Модульні тести (Unit tests). Інтеграційні тести (Integration tests). Функціональні тести (Functional tests). Тести відмови (Regression tests). Тести навантаження (Load tests). Тести безпеки (Security tests). Тести на користувальницький інтерфейс (UI tests). Верифікація програм.

### Тема 12. Маркетинг програмних продуктів.

Дослідження ринку та цільової аудиторії, визначення їхніх потреб, вимог і конкурентного середовища. Створення стратегії продукту. Визначення унікальних пропозицій продукту (Unique Selling Proposition - USP) та ключових переваг, які роблять продукт привабливим для споживачів. Розробка стратегії позиціонування, яка визначає місце продукту на ринку порівняно з конкурентами. Канали просування: Вибір оптимальних каналів для реклами. Створення контенту. залучення користувачів та отримання відгуків. Взаємодія з користувачами, залучення до співпраці та отримання відгуків для покращення продукту та його просування на ринку. Аналіз та оптимізація стратегій.

#### Перелік навчальних робіт студентів та оцінки їх у балах з дисципліни «Технології розподілених систем та паралельних обчислень»

Види робіт	К-сть балів
Лабораторне заняття №1. Тема: «Види програмного забезпечення»	4
Лабораторне заняття №2. Тема: «Життєвий цикл і етапи розробки програмного забезпечення»	4
Лабораторне заняття №3. Тема: «Стандартизація розробки програмного забезпечення»	4
Лабораторне заняття №4. Тема: «Сучасні методології розроблення програмних продуктів»	4
Лабораторне заняття №5. Тема: «Архітектура програмного забезпечення»	4
Лабораторне заняття №6. Тема: «Розподілені архітектури програмного забезпечення»	4
Лабораторне заняття №7. Тема: «Стандарти розробки front-end»	4
Лабораторне заняття №8. Тема: «Стандарти розробки backend»	4
Лабораторне заняття №9. Тема: «Контейнеризація»	4
Лабораторне заняття №10. Тема: «Система керування версіями»	4
Лабораторне заняття №11. Тема: «Якість програмного забезпечення»	4
Лабораторне заняття №12. Тема: «Маркетинг програмних продуктів»	6
Модульний контроль	20
Виконання індивідуального завдання (СР)	30
<b>Разом: Аудиторна робота</b>	<b>70</b>
<b>Самостійна робота (СР)</b>	<b>30</b>
<b>Всього:</b>	<b>100</b>

## КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

При вивченні дисципліни використовуються наступні форми контролю знань студентів: поточний; модульний; підсумковий.

**Поточний контроль** передбачає перевірку теоретичних питань, самостійної роботи, практичних робіт та усне опитування по кожній практичній роботі. По даному виду контролю оцінювання знань здійснюється у відповідності до бального розподілу наведеного в попередній таблиці.

**Модульний контроль** передбачає виконання модульної контрольної роботи. Всі завдання оцінюються в 20 балів. Перше завдання (теоретичне) – 4 бали, друге завдання (практичне) – 8 балів, третє завдання (практичне) – 8 балів.

**Формою підсумкового контролю** є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання двох теоретичних питань (2 x 20 балів = 40 балів) та практичного завдання (60 балів).

**Результуюча оцінка з дисципліни** визначається як середня від балів набраних протягом семестру та отриманих на іспиті.

## 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.