

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

« 29 » червня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Неперервна інтеграція та розгортання ПЗ»

для здобувачів вищої освіти

галузь знань
спеціальність
освітній рівень
освітня програма
вид дисципліни

**F Інформаційні технології
F2 Інженерія програмного забезпечення
бакалавр
Інженерія програмного забезпечення
обов'язкова**

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	4
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: Тимчук Олег Сергійович, кандидат технічних наук, доцент

Пролонговано: на 2027/2028 н.р. _____ (підпис) _____ (ПІБ) « ____ » _____ 2027 р.
(дата)
на 2028/2029 н.р. _____ (підпис) _____ (ПІБ) « ____ » _____ 2028 р.
(дата)
на 2029/2030 н.р. _____ (підпис) _____ (ПІБ) « ____ » _____ 2029 р.
(дата)

Розробник: **Тимчук О.С.**, к.т.н., доцент, доцент кафедри програмних систем і технологій



ЗАТВЕРДЖЕНО

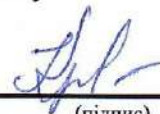
Завідувач кафедри
програмних систем і технологій

 проф. Олексій БИЧКОВ
(підпис)

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії  Ганна КРАСОВСЬКА
(підпис)

© _____, 2026 рік
© _____, 20__ рік
© _____, 20__ рік



1. Мета дисципліни

Метою дисципліни є вивчення принципів, методів і інструментів Pipeline CI/CD (неперервна інтеграція та розгортання) на прикладі різних типів Python-застосунків: console, game, web.

2. Попередні вимоги до опанування або вибору навчальної дисципліни

1. Знати основні поняття та особливості процедурного та об'єктно-орієнтованого програмування.
2. Знати основи створення та обробки структур даних.

3. Анотація навчальної дисципліни

Дисципліна «Неперервна інтеграція та розгортання ПЗ» зосереджена на навчанні технологій та практичних навичок розробки програмного забезпечення. У дисципліні розглядаються принципи CI/CD, їх впровадження та використання для автоматизації процесів розробки та релізу програмного забезпечення.

Здобувачі отримують знання та практичні навички роботи з Git та GitHub, включаючи створення репозиторіїв, гілок та pull request'ів, а також використання GitHub Actions для автоматизації тестування та розгортання проєктів.

Крім того, здобувачі ознайомляться з фреймворком Django та розробкою веб-додатків на його основі, включаючи налаштування бази даних, роботу з шаблонами та налаштування Pipeline CI/CD з GitHub Actions для збірки, тестування та розгортання Django-застосунку на Heroku.

4. Завдання вивчення дисципліни.

В результаті вивчення дисципліни здобувачі набувають знання основних принципів, методів і інструментів неперервної інтеграції та розгортання програмного забезпечення та зможуть застосовувати здобуті знання для покращення процесів розробки командних проєктів.

5. Результати навчання за дисципліною

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та / або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання ¹ (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати основні концепції Continuous Integration/Continuous Deployment (CI/CD) та їх взаємозв'язок з Git та GitHub	лекція	бліц опитування, письмові модульні контрольні роботи, іспит	2%
1.2	Знати мову програмування Python	лекція		2%
1.3	Знати принципи системи контролю версій Git, включаючи команди для роботи з репозиторієм	лекція		4%
1.4	Знати принципи модульного тестування Python-застосунків	лекція		5%
1.5	Знання GitHub Flow та навички роботи зі стандартними робочими процесами	лекція		4%
1.6	Знати формати серіалізації	лекція		4%

1.7	Знати основні концепції та можливості сервісу GitHub Actions	лекція		4%
1.8	Знати основи розробки веб-застосунків на базі Django	лекція		4%
1.9	Знати принципи модульного тестування Django-застосунків	лекція		4%
1.10	Знати принципи розгортання веб-додатків на платформі Heroku	лекція		5%
2.1	Вміти створювати Python-застосунки	лабораторне заняття	бліц опитування, оцінювання лабораторних робіт, виконання додаткових індивідуальних завдань, іспит	до 32%
2.2	Вміти використовувати базові команди Git: init, add, commit, branch, checkout, merge, pull, push	лабораторне заняття		
2.3	Вміти працювати з GitHub: створювати репозиторії, робити коміти, створювати та редагувати гілки, робити pull requests, вирішувати конфлікти злиття гілок	лабораторне заняття		
2.4	Вміти створювати та виконувати unit-тести для Python-застосунків за допомогою pytest	лабораторне заняття		
2.5	Вміти створювати веб-застосунки на базі Django	лабораторне заняття		
2.6	Вміти створювати та виконувати unit-тести для Django-застосунків	лабораторне заняття		
2.7	Вміти працювати з форматами даних JSON та YAML для конфігурації GitHub Actions	лабораторне заняття		
2.8.	Вміти розгортати веб-застосунки на платформі Heroku	лабораторне заняття		
2.9.	Вміти налаштовувати Pipeline CI/CD з GitHub Actions для збірки, тестування та розгортання Django-застосунку на Heroku	лабораторне заняття		
3	комунікація: здатність працювати в команді, вироблення у студентів практичних навиків проведення дослідження і оптимального групового вирішення актуальних проблем із застосуванням відповідних методів і прийомів.	лабораторне заняття, аналітичне дослідження, дискусія, вирішення конкретних задач та ситуацій	виконання творчих командних проектів під час лабораторних робіт	до 10%
4	автономність: продемонструвати розуміння особистої відповідальності за професійні та/або управлінські рішення.			до 20%

Набуття загальних компетентностей (ЗК):

ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК-2. Здатність застосовувати знання у практичних ситуаціях.

ЗК-4. Здатність спілкуватися іноземною мовою як усно, так і письмово.

ЗК-6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК-7. Здатність працювати в команді.

ЗК-13. Здатність використовувати сучасні методології управління проектною розробкою програмного забезпечення.

Набуття фахових компетентностей (ФК):

ФК-1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК-2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК-4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

ФК-6. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки.

ФК-8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК-12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.

ФК-13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни (код)	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	4
Програмні результати навчання (назва)	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9		
ПРН-26.1. Знати архітектуру програмного забезпечення та принципи проектування, володіти технологіями та методологіями розробки та реалізації програмного забезпечення та мережними технологіями, мати практичні навички з адміністрування комп'ютерних мереж в процесі розробки, розгортання та супроводження програмного забезпечення, розуміти і застосовувати програмне забезпечення мережевого обміну, обміну з сенсорами, віддаленої та розподіленої обробки інформації в інтересах інтернету речей.	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

7. Схема формування оцінки

Форми оцінювання: рівень досягнення всіх запланованих результатів навчання визначається за результатами написання письмових контрольних робіт.

Питома вага результатів навчання у підсумковій оцінці за умови її опанування на належному рівні така:

- ✓ результати навчання – **1.1 – 1.10 (знання)** – до 38%;
- ✓ результати навчання – **2 (вміння)** – до 32%;
- ✓ результати навчання – **3 (комунікація)** – до 10%;
- ✓ результати навчання – **4 (автономність)** – до 20%;

7.1. Організація оцінювання:

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності студентів оцінюються за 100-бальною шкалою. Робота в семестрі поділяється на два змістових модуля. Підсумковий контроль (екзамен) проводиться у формі письмової роботи – 40 балів.

У змістовий модуль 1 (ЗМ1) входять теми 1-8 а у змістовий модуль 2 (ЗМ2) – теми 9-17. Обов'язковим для іспиту є виконання та захист студентом лабораторних робіт, передбачених робочою програмою навчальної дисципліни, здача модульних контрольних робіт.

Оцінювання за формами контролю:

Форма контролю	ЗМ1		ЗМ2	
	Мін. – 18 балів	Макс. – 30 балів	Мін. – 18 балів	Макс. – 30 балів
Виконання та захист лабораторних робіт	12	20	12	20
Модульна контрольна робота	6	10	6	10

Поточне семестрове оцінювання:

- оцінюванню підлягають дві модульні контрольні роботи, звіти та захист лабораторних робіт;

- упродовж семестру, після завершення лекційного матеріалу по змістовному модулю 1, орієнтовно на 9 тижні навчання, проводиться письмова модульна контрольна робота 1 із відкритими питаннями.

- упродовж семестру, після завершення лекційного матеріалу по змістовному модулю 2, орієнтовно на останньому тижні навчання, проводиться модульна контрольна робота 2 із відкритими питаннями.

Кожна лабораторна робота (всього чотири) оцінюється в 10 балів, якщо всі завдання виконано, продемонстровано їх працездатність, звіт оформлено відповідно до вимог. Мінімальна кількість балів за лабораторну роботу – 6 балів. Здача всіх лабораторних робіт для отримання допуску до іспиту є обов'язковою умовою.

Розрахунковий мінімум для допуску до іспиту – 36 балів. Критично-розрахунковий мінімум для допуску до іспиту є 20 балів.

Для студентів, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум (20 балів), для допуску до іспиту обов'язковими є виконання та захист запланованих лабораторних робіт і написання хоча б 1 модульної контрольної роботи.

Підсумкове оцінювання (у формі екзамену):

- екзаменаційний білет складається з 20 тестових запитань;

- максимальна кількість балів які можуть бути отримані студентом на іспиті - 40 балів;

- студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів;

- для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен не може бути меншою 24 бали (або 40 балів, якщо допуск було отримано за критично-допустимим мінімумом).

7.2. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

7.3 Політика щодо автоматичної генерації коду

Використання інструментів автоматичної генерації коду в процесі виконання командного проєкту є дозволеним за умови, що студент

1. декларує застосування таких інструментів;
2. демонструє розуміння принципів роботи використаних інструментів, їх місця, переваг та обмежень у контексті проєкту та можливих альтернатив їх застосуванню, що впливають з раніше пройденого матеріалу даної дисципліни та попередніх дисциплін;
3. демонструє розуміння змісту згенерованих фрагментів, способів перевірки їх якості та можливих шляхів їх застосування.

8. Структура навчальної дисципліни

Тематичний план лекцій і лабораторних занять

№ п/ п	Назва лекції	Кількість годин		
		Лекції	Лабораторні	СР
Змістовий модуль 1. Принципи і методи зв'язування даних				
1.	Тема 1. Вступ до CI/CD	2		
2.	Тема 2. Вступ до мови Python. Модель даних у Python. Вирази та інструкції в Python	2	2	2
3.	Тема 3. Функції у Python	2	2	2
4.	Тема 4. Класи у Python	2	2	2
5.	Тема 5. Система управління версіями Git: основні концепції, команди	2	2	4
6.	Тема 6. Контроль версій та спільна робота над проєктом з використанням Git та GitHub	2	2	4
7.	Тема 7. Модульне тестування Python-застосунків	2	2	4
8.	Тема 8. GitHub Flow та робочі процеси	2	2	4
9.	Модульна контрольна робота 1		2	6
Змістовий модуль 2. Принципи і методи обробки пов'язаних даних				
10.	Тема 9. GitHub Actions: огляд сервісу та основні можливості	2	2	4
11.	Тема 10. Серіалізація: JSON, YAML. Використання форматів даних JSON та YAML для конфігурації GitHub Actions	2	2	4
12.	Тема 11. Збірка, тестування та розгортання Python-проєкту на базі GitHub Actions	2	2	4

13.	Тема 12. Розробка веб-застосунків на базі Django: основи та структура проекту	2	3	6
14.	Тема 13. Розробка веб-застосунків на базі Django: диспетчерізація URL-адрес, бази даних, шаблони	2	3	6
15.	Тема 14. Модульне тестування Django-застосунків	2	3	6
16.	Тема 15. Робота з конфігураційними файлами Django-застосунків	2	2	4
17.	Тема 16. Розгортання веб-додатків на Heroku: основні концепції та підходи	2	2	4
18.	Тема 17. Налаштування Pipeline CI/CD з GitHub Actions для збірки, тестування та розгортання Django-застосунку на Heroku	2	3	4
19.	Модульна контрольна робота 2		2	6
	ВСЬОГО	34	40	76

Загальний обсяг 180 год., в тому числі:

Лекції – 34 год.;
 Лабораторні заняття – 40 год.;
 Самостійна робота – 76 год.

9. Рекомендовані джерела:

Основна:

1. Dave Farley. Continuous Delivery Pipelines: How To Build Better Software Faster, 2021 - 160 p.
2. Moritz Lenz. Python Continuous Integration and Delivery. A Concise Guide with Examples, 2019 - 194 p.
3. Jennifer Davis. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, 2016 - 410 p.

Додаткові інформаційні ресурси:

1. Git. – Режим доступу: <https://git-scm.com/>
2. GitHub. – Режим доступу: <https://github.com/>
3. GitHub Actions. – Режим доступу: <https://docs.github.com/en/actions>
4. Офіційний сайт Python. – Режим доступу: <https://www.python.org/>
5. Офіційний сайт Pytest. – Режим доступу: <https://docs.pytest.org/en/7.1.x/contents.html>
6. Офіційний сайт Django. – Режим доступу: <https://www.djangoproject.com/>
7. Документація Heroku. – Режим доступу: <https://devcenter.heroku.com/categories/reference>