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

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

Кафедра математичного моделювання
(назва кафедри, що забезпечує викладання)

"ЗАТВЕРДЖУЮ"
Декан факультету
математики та інформатики
_____ Ольга МАРТИНЮК
"28" серпня 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни
Конвеєри даних та хмарна оркестрація
(назва навчальної дисципліни)

вибіркова
(вказати: обов'язкова)

Освітньо-професійна програма Комп'ютерні науки та проєктування програмних систем
(назва програми)

Спеціальність F3 Комп'ютерні науки
(вказати: код, назва)

Галузь знань F Інформаційні технології
(вказати: шифр, назва)

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

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

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці 2026 рік

Робоча програма навчальної дисципліни «*Конвеєри даних та хмарна оркестрація*» складена відповідно до освітньо-професійної програми «Комп'ютерні науки та проєктування програмних систем»

Розробник:
Косович Ігор Тарасович, асистент кафедри математичного моделювання, доктор філософії за спеціальністю 113 «Прикладна математика»

Викладач, що забезпечує читання даної навчальної дисципліни:
Косович Ігор Тарасович, асистент кафедри математичного моделювання, доктор філософії за спеціальністю 113 «Прикладна математика»

Затверджено на засіданні кафедри математичного моделювання
Протокол № 1 від «27» серпня 2026 року

Завідувач кафедри _____ **Ігор ЧЕРЕВКО**

Схвалено методичною радою факультету математики та інформатики
Протокол № 1 від «28» серпня 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**

Мета навчальної дисципліни: сформувати практичне розуміння архітектури data pipelines та навчити студентів самостійно проєктувати, реалізовувати, перевіряти й оркеструвати процеси збору, зберігання та обробки даних із використанням хмарних технологій.

Пререквізити. Навчальні дисципліни: “Програмування мовою Python”, “Бібліотеки мови Python”.

Результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати: архітектуру data pipelines (ETL/ELT, batch/streaming), протокол HTTP та принципи роботи REST API, методи збору даних з API та вебсайтів (requests, BeautifulSoup, Playwright, Scrapy), принципи побудови надійних data collectors (configuration, logging, retries, validation, incremental collection, deduplication), формати та сховища даних (CSV, JSON, Parquet, PostgreSQL, Google Cloud Storage, BigQuery), підходи до трансформації та перевірки якості даних, основи оркестрації процесів в Apache Airflow;

вміти: аналізувати HTTP-трафік за допомогою Postman, Browser DevTools та Burp Suite; розробляти API collectors та web scrapers; проєктувати підтримувані й відтворювані data collectors; зберігати та трансформувати дані в локальних і хмарних сховищах; виконувати перевірки Data Quality; створювати DAG-и в Apache Airflow та будувати end-to-end data pipeline.

Дисципліна формує такі **компетенції** у відповідності до стандарту вищої освіти спеціальності F3 – Комп'ютерні науки та освітньо-професійної програми:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК5. Здатність вчитися й оволодівати сучасними знаннями.

ЗК9. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК04. Здатність збирати і аналізувати дані (включно з великими), для забезпечення якості прийняття проєктних рішень.

СК05. Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп’ютерних систем різного призначення.

СК07. Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.

СК09. Здатність розробляти та адмініструвати бази даних та знань.

СК11. Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп’ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими **програмними результатами навчання:**

РН9. Розробляти алгоритмічне та програмне забезпечення для аналізу даних (включно з великими).

РН10. Проектувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.

РН12. Проєктувати та супроводжувати бази даних та знань.

РН13. Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.

Опис навчальної дисципліни
Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	1	2	3	90	14	16	–	–	60	–	залік
Заочна											

Структурний зміст навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Теми лекційних занять	**МОДУЛЬ 1. ЗБІР ДАНИХ: API, ВЕБСКРЕЙПІНГ ТА DATA COLLECTORS**											
Тема 1.1. (Лекція) **Вступ до Data Pipelines.**	5	2				3						
Тема 1.2. (Лекція) **HTTP, REST API та аналіз вебтрафіку.**	5	2				3						
Тема 1.3. (Лекція) **Web scraping та advanced data collection.**	5	2				3						
Тема 1.4. (Лекція) **Архітектура data collectors. Надійний та інкрементальний збір даних.**	5	2				3						
Тема 1.5. (Практичне заняття) **Робота з API: від Postman до Python collector.**	6		2			4						

Тема 1.6. (Практичне заняття) **DevTools, undocumented API та Burp Suite.**	6		2			4						
Тема 1.7. (Практичне заняття) **Web scraping: BeautifulSoup, Playwright, Scrapy.**	6		2			4						
Тема 1.8. (Практичне заняття) **Scraping Investigation Challenge.**	6		2			4						
Тема 1.9. (Практичне заняття) **Рефакторинг та інкрементальний collector.**	6		2			4						
Разом за змістовим модулем 1	**50**	**8**	**10**			**32**						
	МОДУЛЬ 2. ЗБЕРІГАННЯ, ТРАНСФОРМАЦІЯ ТА ОРКЕСТРАЦІЯ ДАНИХ											
Тема 2.1. (Лекція) **Зберігання даних: PostgreSQL, Cloud Storage та BigQuery.**	5	2				3						
Тема 2.2. (Лекція) **Трансформація даних та Data Quality.**	5	2				3						
Тема 2.3. (Лекція) **Оркестрація процесів з Apache Airflow.**	5	2				3						
Тема 2.4. (Практичне заняття) **Зберігання даних: PostgreSQL, GCS, BigQuery.**	8		2			6						
Тема 2.5. (Практичне заняття) **Трансформація даних та Data Quality.**	8		2			6						
Тема 2.6. (Практичне заняття) **Повний pipeline в Apache Airflow.**	9		2			7						
Разом за змістовим модулем 2	**40**	**6**	**6**			**28**						
ВСЬОГО	**90**	**14**	**16**			**60**						

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1	**Тема 1.1. Вступ до Data Pipelines.** Що таке data pipeline і навіщо він потрібен. Компоненти: Source → Ingestion → Transformation → Storage → Consumption. ETL та ELT. Batch та streaming pipelines. Джерела даних (API, databases, files, websites) та формати (CSV, JSON, Parquet). Data Lake та Data Warehouse. Принципи надійного pipeline: retries, logging, error handling, idempotency, observability.
2	**Тема 1.2. HTTP, REST API та аналіз вебтрафіку.** HTTP request/response; методи GET, POST, PUT, DELETE; status codes, headers, query parameters, request body. JSON, cookies, sessions. Автентифікація API (API Key, Bearer Token). Pagination та rate limits. REST API. Browser DevTools (Network, Fetch/XHR); пошук internal/undocumented endpoints; Copy as cURL. Burp Suite: Proxy, HTTP History, Repeater.
3	**Тема 1.3. Web scraping та advanced data collection.** HTML, DOM, CSS selectors. Static vs dynamic websites. requests + BeautifulSoup; Playwright для JavaScript-rendered сторінок. Архітектура Scrapy: Spider, Requests, Selectors, Items, Pipelines, Middleware, Settings. robots.txt, умови використання сайтів, відповідальний збір даних. Причини збоїв collectors; DNS, CDN, WAF, origin infrastructure; historical infrastructure research. Anti-bot системи та CAPTCHA; коректні стратегії: official APIs, зменшення collection rate, human-in-the-loop.
4	**Тема 1.4. Архітектура data collectors. Надійний та інкрементальний збір даних.** Script vs data collector. Структура Python-проєкту, separation of concerns. Configuration, .env, secrets management. Logging, error handling, timeouts, retries. Validation отриманих даних. Full vs incremental load; checkpoints; timestamps та unique IDs; deduplication та upsert; exponential backoff; recovery after failure.
5	**Тема 2.1. Зберігання даних: PostgreSQL, Cloud Storage та BigQuery.** CSV, JSON, Parquet: сценарії використання. Database vs files. Реляційні БД та PostgreSQL: schema, table, primary key, index. SQL у контексті data pipelines: SELECT, WHERE, JOIN, GROUP BY, INSERT, UPDATE, UPSERT. Object storage, Data Lake, Data Warehouse. Google Cloud Storage (buckets, objects) та BigQuery (datasets, tables, schema, partitioning). Service accounts. Шари даних RAW → STAGING → CLEAN → ANALYTICS.
6	**Тема 2.2. Трансформація даних та Data Quality.** Transform layer у data pipeline. Трансформації засобами Pandas та SQL: cleaning, типи даних, missing values, duplicates, normalization, aggregation. Data Quality checks: schema, nulls, duplicates, row count, unique IDs, allowed values, ranges. Огляд dbt та PySpark.
7	**Тема 2.3. Оркестрація процесів з Apache Airflow.** Навіщо потрібна workflow orchestration. Apache Airflow: DAG, Task, Operator, TaskFlow API. Dependencies, Scheduler, Cron. Retries, logs, task states. Variables та Connections. XCom, Sensors, branching, catchup та backfill, DAG parameters. Idempotent tasks та best practices.

Детальні презентації до кожної лекції наведено на сайті електронного навчання на сторінці курсу https://moodle.chnu.edu.ua.

Тематика практичних занять (воркшопів) з переліком завдань

№	Назва теми (завдання)
1	**Тема 1.5. Робота з API: від Postman до Python collector.** Робота з реальним API в Postman (Params, Headers, Authorization, Body). Перенесення запиту в Python requests. API collector з pagination; збереження результатів у JSON/CSV.
2	**Тема 1.6. DevTools, undocumented API та Burp Suite.** Пошук XHR-запиту на dynamic website; аналіз параметрів і response. Copy as cURL → Postman → Python. Collector без HTML parsing. Повторення та аналіз запиту через Burp Suite.

3	**Тема 1.7. Web scraping: BeautifulSoup, Playwright, Scrapy.** Scraper на BeautifulSoup з pagination. Збір даних з dynamic page через Playwright. Перший Scrapy Spider; експорт у JSON/CSV.
4	**Тема 1.8. Scraping Investigation Challenge.** Дослідження ресурсу в навчальному/дозволеному середовищі: HTML, Network, API endpoints; аналіз request та pagination; побудова collector. Debugging сценарію, коли collector перестав працювати; стратегія fallback.
5	**Тема 1.9. Рефакторинг та інкрементальний collector.** Розбиття коду collector на модулі; configuration та винесення credentials з коду; logging, exception handling, retry logic. Flow Fetch → Parse → Validate → Transform → Save. Checkpoint, resume after failure, deduplication.
6	**Тема 2.4. Зберігання даних: PostgreSQL, GCS, BigQuery.** Збереження даних у JSON/Parquet. Створення таблиці в PostgreSQL, завантаження даних, SQL queries, basic upsert. Завантаження raw-даних у GCS; створення dataset і таблиці в BigQuery; flow Collector → GCS → BigQuery.
7	**Тема 2.5. Трансформація даних та Data Quality.** Очищення та трансформація RAW dataset засобами Pandas/SQL. Data Quality checks. Формування clean / analytics-ready dataset та завантаження в BigQuery.
8	**Тема 2.6. Повний pipeline в Apache Airflow.** Перший DAG: tasks, dependencies, ручний запуск, logs, retry після штучної помилки. Оркестрація повного pipeline за розкладом: collector → validation → transformation → GCS → BigQuery.

Детальні завдання до кожної практичної роботи наведено на сайті електронного навчання на сторінці курсу
https://moodle.chnu.edu.ua.

Індивідуальні науково-дослідні завдання (ІНДЗ)

№	Завдання до тем
1	Студент може індивідуально виконувати додаткові завдання навчально-дослідницької спрямованості за завданнями, наданими викладачем.
2	Студенти можуть отримати до 10 балів в рахунок ІНДЗ, якщо самостійно пройдуть онлайн-курс з Data Engineering, хмарних технологій (Google Cloud) або Apache Airflow на платформах Prometheus, Coursera чи Google Cloud Skills Boost, отримають відповідний сертифікат і надішлють його на сайт дистанційного навчання викладачу разом з детальним звітом з практичних завдань пройденого курсу (постановки задач, коди виконаних програм, пояснення коду). Кількість балів буде виставлена пропорційно до навчальних результатів студента.

* ІНДЗ – для змістового модуля, або в цілому для навчальної дисципліни визначається викладачем, з урахуванням специфіки дисципліни.

Завдання для самостійної роботи студентів

Самостійна робота складається з повторення матеріалу, засвоєного на лекціях, самостійного опанування частини теоретичного матеріалу, роботи з контрольними запитаннями та завданнями.

Студенти можуть отримувати до 1 бала в рахунок виконання завдань СРС під час кожного лекційного заняття за правильні відповіді на запитання лектора, активне обговорення багатоваріантних підходів до рішення представленої лектором проблеми (для активізації пошукової та дослідної діяльності студентів).

№	Назва теми	Завдання для самостійної роботи	К-сть год.
1	Лекції 1-7	підготовка до лекційних занять	21
2	Практичні 1-8	підготовка до практичних занять (воркшопів), доопрацювання практичних робіт	25
3	Теми 1-15	виконання фінального проєкту та підготовка до його захисту	14

Методи навчання

Методи навчання та викладання: лекції з елементами live coding, практичні заняття у форматі воркшопів (hands-on завдання на реальних або навчальних API, вебсторінках та datasets, debugging sessions, code review), наскрізний проєкт, який розширюється протягом курсу, електронне навчання з використанням системи Moodle, виконання завдань ІНДЗ.

Система контролю та оцінювання

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

Мінімальний пороговий рівень оцінки варто визначати за допомогою якісних критеріїв і трансформувати його в мінімальну позитивну оцінку використовуваної числової (рейтингової) шкали.

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS та є накопичувальною. Протягом семестру студенти виконують 8 практичних робіт (воркшопів). Кожна практична робота оцінюється від 7 до 8 балів; загалом за практичні роботи можна отримати до 60 балів.

Виконуючи завдання практичної роботи, студент повинен завантажити для подальшої перевірки на сайт електронного навчання посилання на репозиторій (GitHub) з працездатним кодом, інструкцією запуску (README) та коротким звітом про виконану роботу.

50% балів, відведених на оцінювання практичної роботи, студент отримує за працюючий програмний продукт: код запускається та відтворює заявлений результат, не містить hardcoded secrets, передбачає базові error handling, logging і retry mechanisms, дані перевіряються перед подальшою обробкою. Решта 50% балів виставляється після захисту роботи, на якому студент має пояснити архітектуру рішення, джерело даних та ключові технічні рішення. При відповіді на теоретичні питання та питання щодо програмної реалізації у випадку неістотної помилки знімається 10-20% балів, а у випадку істотної 20-40% балів; якщо ж студент не опанував теоретичний матеріал, плутається в означеннях,

наводить логічно невірні твердження, то знімається до 50% балів від усієї суми балів за роботу.

Підсумковий контроль проводиться у формі захисту фінального проєкту – end-to-end data pipeline. Максимальна кількість балів, яку можна набрати на захисті проєкту, – 40 балів.

Фінальний проєкт має містити: щонайменше одне реальне або навчальне джерело даних (API / website / file); Python collector або Scrapy spider; validation та базові перевірки Data Quality; збереження raw data; трансформацію даних; завантаження в PostgreSQL та/або GCS / BigQuery; Apache Airflow DAG для scheduling та orchestration; logging, retries та можливість повторного запуску без неконтрольованого дублювання даних.

Критерії оцінювання фінального проєкту: працездатність і відтворюваність pipeline – до 15 балів; архітектура та якість коду (структура, конфігурація, logging, error handling, відсутність hardcoded secrets) – до 10 балів; коректність трансформацій і перевірок Data Quality – до 5 балів; оркестрація в Apache Airflow (DAG, dependencies, retries, scheduling) – до 5 балів; презентація проєкту та відповіді на запитання під час захисту – до 5 балів.

Підсумкова оцінка виставляється за результатами суми балів, набраних на змістових модулях під час семестру та підсумковому модулі згідно з нижченаведеною таблицею.

Методи формування професійної компетентності: розповідь, пояснення, демонстрація, live coding, візуалізація архітектур, дискусія, code review тощо. Методи формування практичних умінь та навичок: виконання завдань воркшопів, розробка data collectors та pipelines, аналіз HTTP-трафіку, debugging, захист практичних робіт і фінального проєкту.

Засобами оцінювання та демонстрування результатів навчання є: програмний код практичних робіт у репозиторії; README та звіти; демонстрація роботи pipeline; презентація та захист фінального проєкту; презентації результатів ІНДЗ; усний контроль у вигляді індивідуального та фронтального опитування на лекціях та практичних заняттях.

Формами поточного контролю є усна чи письмова (практична робота, ІНДЗ) відповідь студента та перевірка програмного коду. Формою підсумкового контролю є залік, який проводиться у формі захисту фінального проєкту.

Критерії оцінювання поточного та підсумкового контролю
Розподіл балів, які отримують студенти

Поточне оцінювання (практичні роботи)								Проєкт	Сума балів
Змістовий модуль №1					Змістовий модуль №2				
Тема 1.5	Тема 1.6	Тема 1.7	Тема 1.8	Тема 1.9	Тема 2.4	Тема 2.5	Тема 2.6		
7	7	8	7	8	8	7	8	40	100

Шкала оцінювання: національна та ЄКТС

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

Перелік питань для самоконтролю та підсумкового контролю навчальних досягнень студентів

1. Поняття data pipeline та його основні компоненти.
2. ETL та ELT: відмінності та сценарії використання.
3. Batch та streaming обробка даних.
4. Формати даних CSV, JSON, Parquet: переваги та недоліки.
5. Data Lake та Data Warehouse.
6. Протокол HTTP: методи, status codes, headers, request body.
7. REST API: автентифікація, pagination, rate limits.
8. Використання Postman та Browser DevTools для аналізу запитів.
9. Пошук та відтворення internal/undocumented API.
10. Призначення Burp Suite (Proxy, HTTP History, Repeater) у налагодженні HTTP-запитів.
11. Web scraping статичних сторінок: requests та BeautifulSoup.
12. Збір даних з динамічних сторінок за допомогою Playwright.
13. Архітектура Scrapy: Spider, Items, Pipelines, Middleware, Settings.
14. Причини збоїв data collectors та підходи до їх діагностики. Anti-bot системи та CAPTCHA.
15. Структура data collector: конфігурація, secrets, logging, retries, validation.
16. Incremental load, checkpoints, deduplication та upsert.
17. PostgreSQL у data pipelines: схема, первинні ключі, індекси, UPSERT.
18. Google Cloud Storage та BigQuery: основні поняття.
19. Шари даних RAW, STAGING, CLEAN, ANALYTICS.
20. Трансформації даних засобами Pandas та SQL.
21. Перевірки Data Quality.
22. Apache Airflow: DAG, Task, Operator, TaskFlow API.
23. Scheduling, retries, Variables та Connections в Apache Airflow.
24. XCom, Sensors, branching, catchup та backfill.
25. Idempotency та відтворюваність data pipelines.

Зарахування результатів неформальної освіти

Зарахування результатів неформальної освіти проводиться у відповідності до "Порядку визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти" (затверджено Вченою радою ЧНУ протокол № 16 від 25 листопада 2024 р.)

https://www.chnu.edu.ua/media/4g5fzssb/poriadok-vyznannia-rezultativ-navchannia-zdobutykh-shliakhom-neformalnoi-ta-abo-informalnoi-osvity.pdf.

Рекомендована література

Основна

1. Reis J., Housley M. Fundamentals of Data Engineering: Plan and Build Robust Data Systems.– O'Reilly Media, 2022.
2. Kleppmann M. Designing Data-Intensive Applications.– O'Reilly Media, 2017.
3. Harenslak B., de Ruiter J. Data Pipelines with Apache Airflow.– Manning Publications, 2021.
4. Mitchell R. Web Scraping with Python: Data Extraction from the Modern Web. 3rd ed.– O'Reilly Media, 2024.
5. Densmore J. Data Pipelines Pocket Reference.– O'Reilly Media, 2021.
6. Apache Airflow Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://airflow.apache.org/docs/.

Допоміжна

7. McKinney W. Python for Data Analysis. 3rd ed.– O'Reilly Media, 2022.
8. Lakshmanan V., Tigani J. Google BigQuery: The Definitive Guide.– O'Reilly Media, 2019.
9. Юрченко І.В., Сікора В.С. Програмування мовою Python: Навч. посібник.– Чернівці: Чернівецький національний університет, 2022.– 104 с. https://archer.chnu.edu.ua/xmlui/handle/123456789/3666
10. Scrapy Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://docs.scrapy.org/.

Інформаційні ресурси

http://moodle.chnu.edu.ua
https://docs.python.org/3/
https://requests.readthedocs.io/
https://playwright.dev/python/
https://www.postgresql.org/docs/
https://cloud.google.com/storage/docs
https://cloud.google.com/bigquery/docs
https://learning.postman.com/docs/
https://portswigger.net/burp/documentation

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- "Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- "Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- "Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha-zi-zminamy-zatverdzhenymy-vchenoiu-radoiu-chnu-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- *допускається* за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ *можуть визначатися* викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.

Здобувач зобов'язаний своєчасно та якісно виконувати всі отримані завдання; за необхідністю з метою з'ясування всіх незрозумілих під час самостійної та індивідуальної роботи питань, відвідувати консультації викладача.

Складання/перескладання екзаменів/заліків відбувається за встановленим деканатом розкладом.