



Національний
транспортний
університет

ВК Реінжиніринг та еволюція програмних систем

Рівень вищої освіти – *другий (магістерський)*

Дні занять, час занять, аудиторія: будуть доступні згідно розкладу за посиланням <https://kizt.ntu.edu.ua/to-the-student/rozklady-zanyat-i-sesij/>

Кафедра інформаційних технологій та штучного інтелекту

Лекційні заняття проводить

к.ф.-м.н., доц. Ткаченко Ольга Іванівна

Контактна інформація	oitkachen@gmail.com +38063-282-43-60
----------------------	--

Адреса, номер аудиторії	м. Київ, вул. Івана Огієнка, 19, каб. 601a
-------------------------	--

Час консультацій	Щовівторка з 16:00 до 17:00
------------------	-----------------------------

Практичні заняття проводить

к.ф.-м.н., доц. Ткаченко Ольга Іванівна

Контактна інформація	oitkachen@gmail.com +38063-282-43-60
----------------------	--

Аудиторія	м. Київ, вул. Івана Огієнка, 19, каб. 601a
-----------	--

Час консультацій	Щовівторка з 16:00 до 17:00
------------------	-----------------------------

Анотація курсу: «Реінжиніринг та еволюція програмних систем» - вибіркова навчальна дисципліна, спрямована на формування інженерних умінь аналізувати спадкові програмні системи, планувати їхню модернізацію та керувати еволюційними змінами. У курсі розглядаються відновлення архітектури, технічний борг, рефакторинг, поетапна міграція, сумісність API і даних, версіювання та доказове оцінювання результатів еволюції.

Практичний фокус дисципліни: інвентаризація спадкового репозиторію, побудова карти залежностей, виявлення технічного боргу, виконання контрольованих рефакторингів, проєктування цільової архітектури, підготовка планів міграції та еволюції API/даних, вимірювання результатів і документування рішень.

Предметом вивчення є методи та інструменти реінжинірингу й еволюції програмних систем: відновлення архітектури, аналіз залежностей, керування технічним боргом, рефакторинг, модернізація компонентів, міграція, еволюція API та даних і оцінювання наслідків змін. Обсяг дисципліни - 4 кредити ЄКТС (120 годин): денна форма - 18 годин лекцій, 18 годин практичних занять і 84 години самостійної роботи; заочна форма - 2 години лекцій, 4 години практичних занять і 114 годин самостійної роботи.

Мета дисципліни: сформувати здатність аналізувати спадкові програмні системи, обґрунтовувати стратегію їхньої модернізації, виконувати безпечні структурні зміни та документувати контрольовану еволюцію програмних компонентів і контрактів.

Очікувані результати навчання: після завершення дисципліни здобувач освіти зможе: 1) визначати чинники еволюції, обмеження, проблемні ділянки й технічний борг спадкової системи; 2) відновлювати архітектурні подання та аналізувати залежності між компонентами; 3) обирати й

виконувати безпечні рефакторинги та оцінювати зміни за метриками; 4) проєктувати цільову архітектуру й поетапну модернізацію компонентів із визначенням меж і точок повернення; 5) розробляти сумісні стратегії еволюції API, схем даних і програмних контрактів із версіюванням та депрекацією; 6) порівнювати стани системи до і після змін, готувати доказовий звіт і рекомендації щодо подальшого супроводу.

Міждисциплінарні зв'язки: дисципліна спирається на результати попереднього навчання з програмування, моделювання, баз даних, операційних систем та проєктування програмного забезпечення.

Програма навчальної дисципліни складається з таких модулів:

Модуль 1. Аналіз спадкових систем та підготовка еволюційних змін

Тема 1. Спадкові програмні системи та чинники їх еволюції.

Питання: ознаки спадкових (legacy) систем; бізнесові й технічні чинники змін; обмеження та зацікавлені сторони; атрибути якості; поняття технічного боргу; виявлення проблемних ділянок і сценаріїв змін; пріоритизація еволюційних потреб. Практичне завдання: для обраного репозиторію скласти інвентаризацію системи, профіль проблем і таблицю ризиків та обмежень. Результат і перевірка: профіль системи й упорядкований перелік еволюційних потреб; оцінюється повнота, обґрунтованість пріоритетів і зв'язок із вихідними даними.

Тема 2. Відновлення архітектури та аналіз залежностей.

Питання: відновлення архітектури за кодом і конфігураціями; статичний та динамічний аналіз; подання компонентів і залежностей; зв'язність, зчеплення та циклічні залежності; архітектурна ерозія; зони ризику; зворотне проєктування. Практичне завдання: побудувати карту компонентів і залежностей спадкової системи за допомогою відповідного інструмента та визначити критичні зв'язки. Результат і перевірка: архітектурне подання з поясненнями й переліком виявлених відхилень; оцінюється трасованість висновків і коректність інтерпретації залежностей.

Модуль 2. Керована трансформація та супровід еволюції

Тема 3. Рефакторинг коду та зменшення технічного боргу.

Питання: ознаки проблемного коду; принципи рефакторингу; малі безпечні кроки та точки розділення; автоматизовані рецепти трансформацій; складність, дублювання та супроводжуваність; перевірка збереження поведінки як засіб безпеки змін; порівняння станів до і після. Практичне завдання: обрати проблемні фрагменти, виконати 2–3 рефакторинги вручну або за допомогою OpenRewrite та порівняти метрики. Результат і перевірка: набір змін із поясненням і таблицею метрик; оцінюється збереження поведінки, доцільність кроків і підтверджене зменшення технічного боргу.

Тема 4. Реструктуризація архітектури та модернізація компонентів.

Питання: модульність і межі компонентів; модульний моноліт; фасад, адаптер та антикорупційний шар; патерн поетапної заміни Strangler Fig; виділення швів для міграції; інкрементальні переходи; критерії готовності та повернення. Практичне завдання: запропонувати цільову архітектуру і поетапну міграцію одного підсистемного фрагмента спадкового продукту. Результат і перевірка: схеми поточного та цільового станів і дорожня карта міграції; оцінюється узгодженість меж, здійсненність кроків та оборотність рішень.

Тема 5. Еволюція API, даних і програмних контрактів.

Питання: сумісність інтерфейсів; версіювання API та схем даних; семантичне версіювання; депрекація; адаптери; стратегія expand-and-contract; перетворення даних; прапорці функцій; зворотна й пряма сумісність; план повернення. Практичне завдання: розробити план сумісної зміни API або схеми даних із матрицею версій і прикладом міграції. Результат і перевірка: версійований контракт, матриця сумісності та скрипт або псевдокод міграції; оцінюється повнота обмежень, керування ризиками та наявність сценарію повернення.

Тема 6. Оцінювання результатів еволюції та керування змінами.

Питання: аналіз впливу змін; відповідність цільовій архітектурі; метрики якості та супроводжуваності; тренд технічного боргу; готовність до випуску; спостережуваність; критерії приймання й повернення; записи архітектурних рішень і супровідна документація. Практичне завдання: порівняти стани системи до і після змін та підготувати звіт з метриками, ризиками, залишковим боргом і критеріями випуску/відкату. Результат і перевірка: підсумковий звіт еволюції; оцінюється доказовість метрик, послідовність висновків і якість управлінських рекомендацій.

Індивідуальне завдання окремо не передбачено освітньою програмою та навчальним планом; самостійні роботи за темами 1–6 виконуються в межах поточного й модульного контролю.

Окремі індивідуальні роботи з обов'язковим оцінюванням не встановлюються; результати самостійної роботи перевіряються в межах поточних практичних і модульних завдань.

Методи поточного контролю: виконання та захист практичних завдань за темами 1–6, інвентаризація репозиторію, побудова карт залежностей, аналіз технічного боргу, виконання рефакторингів, підготовка міграційних планів, еволюція контрактів і обґрунтування інженерних висновків.

Методи модульного контролю: для денної форми - дві модульні контрольні роботи по 15 балів; для заочної форми - контрольна робота на 30 балів. Підсумковий контроль - екзамен, 40 балів.

Джерела для вивчення дисципліни - рекомендовані навчальні матеріали, стандарти та офіційна документація наведені в кінці силабусу.

1. Практичні матеріали курсу: зразки спадкових репозиторіїв, карти залежностей, рефакторингові рецепти, шаблони міграційних планів, контракти API/даних і звіти з метриками, що надаються викладачем.

2. Електронний ресурс бібліотеки НТУ: <http://lib.ntu.edu.ua/catalog/login.html>

3. Віртуальне середовище навчання (Google Classroom): посилання на курс надається викладачем на першому занятті.

4. Основні програмні засоби: OpenRewrite, SonarQube, ArchUnit, Git, засоби статичного аналізу та візуалізації залежностей; конкретні версії й спосіб установлення визначаються під час виконання практичних робіт.

Оцінювання

Контроль протягом семестру (60 балів)					Екзамен	Сума балів
Модуль 1		Модуль 2		Модуль 3 (ІЗ)		
Тема 1	Тема 2	Тема 3	Теми 4–6			
Для денної форми здобуття вищої освіти: - практичні завдання за темами 1–6 - по 5 балів (усього 30); - модульна контрольна робота № 1 - 15 балів; - модульна контрольна робота № 2 - 15 балів; - поточний контроль разом - 60 балів.					40	100
Для заочної форми здобуття вищої освіти: - письмові/дистанційні завдання за темами 1–6 - по 5 балів (усього 30); - контрольна робота - 30 балів; - поточний контроль разом - 60 балів.						

Самостійна робота

Самостійна робота: опрацювання літератури, стандартів і документації, аналіз спадкових репозиторіїв, відновлення архітектури, підготовка рефакторингів і міграційних планів, дослідження сумісності API/даних, обчислення метрик та оформлення технічних висновків. Результати самостійної роботи перевіряються в межах поточних практичних і модульних завдань.

Завдання для самостійної роботи виконуються на матеріалах практичних робіт і обговорюються під час консультацій; їхні результати враховуються в поточному та модульному контролі.

Результати навчання, здобуті у неформальній або інформальній освіті, можуть бути визнані за умови їх відповідності конкретним результатам навчання дисципліни та проходження процедури, визначеної чинним положенням НТУ. Визнані результати зараховуються замість відповідних завдань у межах 60 балів поточного та модульного контролю й не збільшують максимальну оцінку понад 100 балів.

Критерії оцінювання

Критерії оцінювання: практичні та контрольні роботи оцінюються за повнотою аналізу спадкової системи, коректністю архітектурних подань і метрик, обґрунтованістю рефакторингів і міграційних кроків, урахуванням сумісності та можливості повернення, доказовістю результатів, якістю звіту та висновків; екзамен - за правильністю виконання завдань і здатністю обґрунтувати рішення.

Політика несвоєчасного проходження контрольних заходів.

Поточний та підсумковий контролі проводяться згідно з графіком освітнього процесу та затвердженими графіками. У разі неявки на підсумковий контроль з поважної причини контроль може бути проведений індивідуально в погоджений строк за наявності дозволу дирекції ННКІЗТ.

Повторне складання екзамену в разі отримання незадовільної оцінки допускається не більше двох разів: один раз - викладачеві, другий - комісії, створеній директором ННКІЗТ відповідно до чинних нормативних документів НТУ.

У разі подання індивідуального завдання пізніше встановленого строку без поважної причини оцінка знижується на 10 %. Документовані збої електронного середовища навчання, аварійні відключення або інші обставини, що об'єктивно унеможливили своєчасне подання, розглядаються викладачем індивідуально.

Політика перескладання. Упродовж тижня після оголошення результатів поточного контролю здобувач освіти може звернутися до викладача за роз'ясненням або з обґрунтованою незгодою. Апеляція щодо результату підсумкового контролю подається в порядку та строки, визначені чинними нормативними документами НТУ. Повторне проходження підсумкового контролю з метою підвищення позитивної оцінки не допускається.

Політика відвідування та активності. Здобувач освіти має брати участь у навчальних заняттях і виконувати завдання, передбачені курсом. Пропущені практичні роботи та контрольні заходи виконуються за погодженням індивідуальним графіком до завершення підсумкового контролю. Рішення про допуск до екзамену ґрунтується на виконанні обов'язкових оцінюваних завдань, а не лише на факті відвідування занять.

Академічна доброчесність.

Дотримання академічної доброчесності регулюється Кодексом етики академічних взаємовідносин та доброчесності НТУ, Положенням про систему забезпечення академічної доброчесності та іншими чинними нормативними документами університету.

Кодекс етики академічних взаємовідносин та доброчесності Національного транспортного університету

Положення про систему забезпечення академічної доброчесності педагогічними, науково-педагогічними, науковими працівниками та здобувачами вищої освіти в Національному транспортному університеті

Порушеннями академічної доброчесності є:

- академічний плагіат;
- фальсифікація;
- списування;
- обман;
- неправомірна вигода;
- хабарництво.

Види порушень академічної доброчесності та відповідальність за них визначаються законодавством України й чинними нормативними документами НТУ. Рішення щодо наявності порушення приймається з урахуванням установленної процедури та обставин конкретної роботи.

Під час поточного або підсумкового контролю дозволено використовувати лише ті джерела й засоби, які прямо визначив викладач. У разі виявлення ознак списування, недозволеної допомоги або підміни авторства результат роботи розглядається відповідно до процедур академічної доброчесності НТУ.

Поведінка в аудиторії. Ноутбуки та портативні пристрої можна використовувати з навчальною метою відповідно до завдань курсу. Не допускається використання пристроїв, яке перешкоджає заняттю або порушує академічну доброчесність. Обмеження не поширюються на погоджені засоби доступності та допоміжні технології.

В аудиторії забороняється вживання їжі, напоїв (за винятком води). Здобувачі вищої освіти та викладачі повинні дотримуватися етичних норм поведінки.

Здобувачам освіти з інвалідністю та/або особливими освітніми потребами рекомендовано до початку семестру звернутися до дирекції ННКІЗТ для погодження необхідних умов і засобів підтримки навчання.

У разі виникнення проблем зі здоров'ям, які можуть перешкоджати навчанню, здобувачеві освіти слід звернутися до медичного закладу та повідомити про обставини дирекцію ННКІЗТ у прийнятний спосіб.

Скарги, пропозиції, зауваження та повідомлення про конфліктні ситуації в межах освітніх програм можна надсилати на електронну адресу general@ntu.edu.ua або через скриньку довіри, розміщену при вході до університету.

Електронна скринька для звернень до психологічної служби: philosophy@ntu.edu.ua.

Зв'язок з викладачем: e-mail: oitkachen@gmail.com; тел.: +38063-282-43-60.

Рекомендована література:

1. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Addison-Wesley, 2018. URL: <https://martinfowler.com/books/refactoring.html> (дата звернення: 20.08.2026).
2. Fowler M. Strangler Fig Application. [Online]. URL: <https://martinfowler.com/bliki/StranglerFigApplication.html> (дата звернення: 20.08.2026).
3. OpenRewrite. Documentation: automated refactoring recipes. [Online]. URL: <https://docs.openrewrite.org/> (дата звернення: 20.08.2026).
4. SonarSource. Understanding measures and metrics: maintainability and technical debt. [Online]. URL: <https://docs.sonarsource.com/sonarqube-server/2025.4/user-guide/code-metrics/metrics-definition> (дата звернення: 20.08.2026).
5. ArchUnit. User Guide. Version 1.5.0. [Online]. URL: https://www.archunit.org/userguide/html/000_Index.html (дата звернення: 20.08.2026).
6. Semantic Versioning. Semantic Versioning 2.0.0. [Online]. URL: <https://semver.org/> (дата звернення: 20.08.2026).
7. Git. Reference documentation. [Online]. URL: <https://git-scm.com/docs> (дата звернення: 20.08.2026).
8. ISO/IEC/IEEE 14764:2022. Software engineering - Software life cycle processes - Maintenance. [Online]. URL: <https://www.iso.org/standard/80710.html> (дата звернення: 20.08.2026).

Додаткова інформація

Детальнішу інформацію щодо методів навчання, практичних занять, форм оцінювання, самостійної роботи та повного списку літератури наведено у Робочій програмі навчальної дисципліни.