

Міністерство освіти і науки України
Тернопільський національний технічний університет
імені Івана Пулюя

ЛІТЕРАТУРА



НАВЧАЛЬНО-МЕТОДИЧНА

Кафедра програмної інженерії

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсового проєкту з дисципліни
«Рациональний уніфікований процес проєктування
програмного забезпечення»
для здобувачів другого (магістерського) рівня вищої освіти
за освітньо-професійною програмою
«Інженерія програмного забезпечення»
спеціальності F2 – «Інженерія програмного забезпечення»
всіх форм навчання

ТЕРНОПІЛЬ
ТНТУ ім. І. Пулюя
2025

Методичні вказівки до виконання курсового проєкту з дисципліни «Раціональний уніфікований процес проєктування програмного забезпечення» розроблені у відповідності з робочою програмою з дисципліни та освітньою програмою «Інженерія програмного забезпечення» другого (магістерського) рівня вищої освіти за спеціальністю F2 «Інженерія програмного забезпечення» галузі знань F «Інформаційні технології»

Укладачі:

Мудрик І.Я., PhD, доцент
Петрик М.Р., д.ф.-м.н., професор
Михалик Д.М., к.т.н., доцент

Рецензент:

Баран І.О., к.т.н, доцент,
Декан факультету комп'ютерно-інформаційних систем та програмної інженерії
ТНТУ ім. Івана Пулюя

Відповідальний за випуск: доц. каф. ПІ Мудрик І.Я.

Методичні вказівки розглянуто й затверджено на засіданні
кафедри програмної інженерії Тернопільського національного технічного університету імені
Івана Пулюя.

Протокол № 1 від 1 вересня 2025 р.

Схвалено та рекомендовано до друку на засіданні науково-методичної комісії
факультету комп'ютерно-інформаційних систем і програмної інженерії
Тернопільського національного технічного університету імені Івана Пулюя.

Протокол № 1 від 2 вересня 2025 р.

Методичні вказівки до виконання курсового проєкту з дисципліни «Раціональний уніфікований процес проєктування програмного забезпечення» для здобувачів ОКП «Магістр» спеціальності F2 – «Інженерія програмного забезпечення», всіх форм навчання / укладачі: Мудрик І.Я., Петрик М.Р., Михалик Д.М. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. – 49 с.

Методичні вказівки призначені для ознайомлення з вимогами, правилами оформлення та оцінювання курсових проєктів для здобувачів вищої освіти ОКП «Магістра», які навчаються за спеціальністю F2 «Інженерія програмного забезпечення». Методичні вказівки розроблено на основі діючого законодавства, стандарту вищої освіти, освітньо-професійної програми «Інженерія програмного забезпечення» другого "магістерського" рівня вищої освіти, робочої програми та силабуса з дисципліни «Раціональний уніфікований процес проєктування програмного забезпечення». В методичних вказівках розкрито суть та загальні положення щодо виконання курсового проєкту; вказані основні вимоги до структури, змісту та оформлення проєкту; наводяться критерії оцінювання. У додатках наведено зразки титульної сторінки, приклад змісту пояснювальної записки та примірну тематику курсових проєктів.

Зміст

ВСТУП	4
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	5
2 МЕТА Й ЗАВДАННЯ КП.....	7
3. ОРГАНІЗАЦІЯ ВИКОНАННЯ КП.....	9
3.1 Підготовчий етап.....	11
3.2 Основний етап	13
3.3 Заключний етап	14
3.5 Права та обов'язки здобувача	14
4 ВИМОГИ ДО ЗМІСТУ ТА СТРУКТУРИ КП	16
5 ВИМОГИ ДО ОФОРМЛЕННЯ.....	19
5.1 Загальні вимоги	19
5.2 Вимоги до обов'язкових структурних елементів КП	22
5.3 Оформлення списків	24
5.4 Оформлення ілюстрацій	25
5.5 Оформлення таблиць	26
5.6 Оформлення формул та рівнянь	27
5.7 Оформлення лістингів програмного коду.....	28
5.8 Оформлення посилань на літературні джерела.....	29
5.9 Оформлення додатків	30
5.10 Оформлення коду та супровідної документації	31
6 ПОРЯДОК ПЕРЕВІРКИ ТА ЗАХИСТУ	31
6.1 Підготовка до захисту.....	31
6.2 Академічна доброчесність.....	33
6.2 Підготовка презентації та доповіді.....	34
6.3 Порядок захисту курсового проєкту	36
6.4 Оцінювання курсових проєктів	37
СПИСОК ДЖЕРЕЛ	39
ДОДАТКИ.....	42
ДОДАТОК А.....	42
ДОДАТОК Б	44
ДОДАТОК В.....	46
ДОДАТОК Г	48
АКТУАЛЬНИЙ ПРИКЛАД ЗМІСТУ КУРСОВОГО ПРОЄКТУ (РІВЕНЬ МАГІСТРА)	48

ВСТУП

Курсовий проєкт із дисципліни «Раціональний уніфікований процес проєктування ПЗ» є ключовим етапом підготовки магістрів спеціальності 121 «Інженерія програмного забезпечення». Він спрямований на інтеграцію теоретичних знань та формування поглиблених практичних компетенцій у сфері стратегічного проєктування складних програмних систем.

Основні завдання курсового проєкту полягають у:

- Синтезі архітектурних рішень: виборі та обґрунтуванні стійких, масштабованих архітектурних стилів і патернів, що відповідають сучасним вимогам до Cloud-native та розподілених систем.
- Використанні методології RUP: адаптації ітеративного підходу до управління життєвим циклом розробки, зокрема дисциплін аналізу, проєктування та управління ризиками.
- Поглибленому моделюванню: деталізації компонентів ПЗ (мікросервісів, підсистем, класів) на основі складних вимог зацікавлених осіб із застосуванням нотації UML 2.5.
- Інструментальному забезпеченні: опануванні сучасних CASE-засобів (Enterprise Architect, Visual Paradigm тощо) та інструментів архітектурного аналізу для автоматизації процесу розробки артефактів.
- Забезпеченні простежуваності (Traceability): встановленні чітких зв'язків між бізнес-цілями, функціональними вимогами та програмною реалізацією.

Основна мета виконання курсового проєкту полягає у демонструванні здобувачем високого рівня оволодіння фаховими **компетентностями** та досягненні програмних **результатів навчання**, визначених освітньо-професійною програмою другого (магістерського) рівня. Проєкт є підсумком практичного застосування знань у сферах стратегічного архітектурного аналізу та детального проєктування складних програмних систем.

В методичних вказівках наведено рекомендації щодо тем, структури, змісту, наповнення та форматування курсових проєктів, а також описано загальний порядок організації процесу та захисту курсових проєктів ОКП «Бакалавр» на кафедрі програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, відповідно до вимог діючих нормативних документів.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Підготовка фахівців другого (магістерського) рівня вищої освіти здійснюється на основі освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю F2 «Інженерія програмного забезпечення». Дана програма передбачає поглиблену науково-практичну та професійно-орієнтовану підготовку, спрямовану на розв'язання складних задач проєктування та управління розробкою інтелектуальних і високотехнологічних програмних систем.

Курсовий проєкт з дисципліни «Раціональний уніфікований процес проєктування ПЗ» для здобувачів другого (магістерського) рівня вищої освіти (надалі – КП) є самостійною оригінальною працею. Він демонструє здатність здобувача розв'язувати комплексні завдання з аналізу, розробки архітектури та багаторівневого проєктування програмних систем, що визначені Стандартом вищої освіти та освітньо-професійною програмою «Інженерія програмного забезпечення» за спеціальністю F2.

З метою перевірки відповідності результатів навчання вимогам освітньої програми проводиться проміжна атестація здобувачів у формі публічного захисту курсового проєкту.

Основним критерієм при виконанні проєкту є методологічний підхід та практична цінність, архітектурна обґрунтованість та практична доцільність обраних рішень. Робота над проєктом базується на засадах методології Rational Unified Process (RUP), що передбачає:

- Ітеративний та інкрементний підхід: поступове зниження ризиків та уточнення архітектури на кожній фазі життєвого циклу.
- Управління вимогами: перехід від бізнес-моделювання до детальних специфікацій (Use Cases, Supplementary Specs).
- Використання компонентної архітектури: проєктування системи як сукупності взаємопов'язаних підсистем та сервісів, що забезпечує високу масштабованість та придатність до супроводу.
- Проєктування та якість коду: Здобувачі мають застосовувати об'єктно-орієнтований аналіз та дизайн (OOAD) для виокремлення абстракцій, що формують стійкий архітектурний каркас. Оцінювання якості проєкту базується на принципах чистої архітектури: мінімізації зв'язності (coupling) та підвищенні згуртованості (cohesion) компонентів. Використання складних архітектурних патернів повинно бути виправданим і спрямованим на підвищення надійності, продуктивності та безпеки ПЗ.

Дисциплінарна інтеграція: У курсовому проєкті з дисципліни «Раціональний уніфікований процес проєктування ПЗ» магістри вирішують

комплексні інженерні задачі, що базуються на компетенціях, здобутих під час вивчення таких дисциплін, як: «Моделювання та аналіз програмного забезпечення»; «Об'єктно-орієнтоване програмування»; «Архітектура програмного забезпечення»; «Управління ІТ-проєктами», «Якість та тестування ПЗ», «Менеджмент проєктів ПЗ».

Курсовий проєкт інтегрує ці знання для створення цілісного набору проєктних артефактів — від моделі бізнес-процесів до фізичної схеми розгортання системи відповідно до сучасних стандартів галузі.

Процес підготовки курсових проєктів на кафедрі програмної інженерії відбувається відповідно до вимог Положення «Про підсумковий семестровий контроль результатів навчання студентів Тернопільського національного технічного університету імені Івана Пулюя» [4, 5], законів України “Про освіту” [2] та “Про вищу освіту” [3], Положень “Про організацію освітнього процесу в Тернопільському національному технічному університеті імені Івана Пулюя” [5], “Про недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя” [6], “Про академічну доброчесність учасників освітнього процесу Тернопільського національного технічного університету імені Івана Пулюя” [7], інших чинних нормативних та розпорядчих документів.

Курсовий проєкт є об'єктом обов'язкової перевірки на академічний плагіат. Робота не може містити ознак фальсифікації, списування або несанкціонованого використання чужих ідей та результатів без належних посилань. Дотримання принципів академічної доброчесності є обов'язковою умовою допуску до захисту.

На підставі результатів публічного захисту екзаменаційна комісія приймає рішення щодо оцінювання курсового проєкту здобувача з дисципліни «Раціональний уніфікований процес проєктування ПЗ».

У даному посібнику наведено систематизовані вимоги до організації процесу виконання курсового проєкту, його внутрішнього змісту, логічної структури та технічного оформлення артефактів згідно з методологією RUP.

2 МЕТА Й ЗАВДАННЯ КП

Метою виконання курсового проєкту є демонстрація здобувачем здатності до комплексного та творчого розв'язання складних завдань проєктування програмного забезпечення прикладного характеру. Робота базується на поглиблених теоретичних знаннях та професійних навичках, що відповідають Стандарту вищої освіти України за спеціальністю F2 «Інженерія програмного забезпечення» та відповідній освітньо-професійній програмі другого (магістерського) рівня.

Основними завданнями курсового проєкту є:

- Систематизація та поглиблення компетенцій: закріплення та розширення теоретичних знань і практичних навичок архітектурного проєктування, отриманих протягом навчання, а також їх критичне застосування при вирішенні конкретних технічних задач розробки високонавантажених або розподілених систем.
- Оволодіння методологією RUP: розвиток навичок самостійної роботи з артефактами раціонального уніфікованого процесу, включаючи дисципліни аналізу вимог, об'єктно-орієнтованого моделювання та управління конфігураціями.
- Архітектурне та детальне проєктування: набуття досвіду у виборі та обґрунтуванні архітектурних стилів, патернів та стратегій декомпозиції ПЗ на підсистеми, компоненти та класи на основі специфікацій вимог.
- Інструментальна майстерність: ефективне використання сучасних CASE-засобів (Computer-Aided Software Engineering), засобів автоматизації моделювання та принципів системного аналізу для створення цілісної моделі програмної системи.
- Професійна аналітика: розвиток здатності аналізувати альтернативні підходи до проєктування, оцінювати ризики та приймати зважені рішення щодо вибору технологічного стеку та структури програмного продукту.

Виконання КП покликане набути здобувачами інтегральної компетентності, а саме, здатності особи розв'язувати складні задачі і проблеми у певній галузі професійної діяльності або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов та вимог.

Загальні компетентності [7]:

- ЗК01 Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК05 Здатність генерувати нові ідеї (креативність).

Спеціальні (фахові) компетентності:

- СК1 Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.
- СК2 Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.
- СК3 Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.
- СК5 Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення .
- СК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.
- СК09. Здатність забезпечувати якість програмного забезпечення.
- СК10*. Здатність розробляти інноваційне програмне забезпечення.
СК* - спеціальна компетентність, запропонована стейкхолдерами.

Курсовий проєкт в рамках зазначеної дисципліни для магістрів усіх форм навчання передбачає формування у здобувачів таких програмних результатів навчання згідно освітньо-професійної програми F2:

- РН01 Знати і застосовувати сучасні професійні стандарти і інші нормативно правові документи з інженерії програмного забезпечення.
- РН02 Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.
- РН03 Будувати і досліджувати моделі інформаційних процесів у прикладній області.
- РН06 Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.
- РН08 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
- РН11 Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.
- РН17 Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

Область професійної діяльності – розробка програмних продуктів, технологій та засобів розроблення програмного забезпечення, наукові дослідження, викладацька, експертна та консультативна діяльність у сфері інженерії програмного забезпечення.

Магістри мають змогу виконувати завдання розробника складних інформаційних систем та програмного забезпечення у різних сферах, архітекторами програмних систем (Software Architect), керівниками команд розробників (Team Lead), розробниками програмного забезпечення (Software Engineer), інженерами із забезпечення якості та тестування програмного забезпечення (QA), розробниками та аналітиками надвеликих сховищ даних, DevOps-інженерами тощо, працевлаштуватися на посади з наступною назвою професійної роботи згідно з Національним класифікатором професій ДК 003:2010:

КП є самостійно виконаною і відповідно оформленою працею здобувача з вказаної дисципліни з вирішенням конкретного завдання проєктування ПЗ на основі набутих у процесі навчання знань та практичних навичок.

3. ОРГАНІЗАЦІЯ ВИКОНАННЯ КП

Формування тематики та змісту завдань:

Завдання на курсовий проєкт має бути орієнтоване на розв’язання фундаментальних та прикладних задач архітектурного проєктування на всіх етапах життєвого циклу ПЗ: від стратегічного аналізу вимог до генерації програмного коду на основі моделей. Магістерський рівень передбачає не лише застосування відомих методологій, а й адаптацію процесів Rational Unified Process (RUP) до специфічних (нестандартних) умов розробки, таких як мікросервісна організація, хмарні обчислення або інтеграція штучного інтелекту.

Джерела формування тематики:

Орієнтовний перелік тем розробляється випусковою кафедрою з урахуванням фокусу освітньо-професійної програми «Інженерія програмного забезпечення» (див. Додаток Г) та досвіду науково-педагогічних працівників (НПП).

Тематика КП тісно пов’язана з актуальними науково-технічними дослідженнями кафедри, професійними інтересами викладачів, а також запитами ІТ-індустрії та партнерських підприємств.

Індивідуалізація тем:

Здобувачі мають право пропонувати власні теми КП, якщо вони корелюють з їхньою науково-дослідною роботою або поточною професійною діяльністю в ІТ-секторі. Такі теми затверджуються за умови згоди відповідного

викладача кафедри на здійснення наукового керівництва та відповідності теми складним інженерним вимогам магістерського рівня.

Вимоги до теми:

Обрана тема повинна бути актуальною, мати наукове обґрунтування та прикладну цінність. Проект має відображати сучасний стан індустрії, використовувати новітні архітектурні патерни та інструментальні засоби автоматизації проектування (CASE-засоби), що відповідають світовим стандартам інженерії програмного забезпечення. Орієнтовний перелік тематики КП пропонується випусковою кафедрою урахуванням специфіки освітньо-професійної програми «Інженерія програмного забезпечення» (див. Додаток Г) [8, 9], досвіду керівництва курсовими проектами кафедри; науково-технічних досліджень та розробок, професійних інтересів науково-педагогічних працівників (НПП) кафедри, замовлень і рекомендацій ЗВО, виробничих підприємств та інших установ і відомств [5].

Окремі теми КП, що можуть бути пов'язані з дослідницькою роботою здобувачів на кафедрі або їх професійною діяльністю, можуть бути запропоновані здобувачами, за умови, що відповідні НПП кафедри дали згоду на керівництво підготовкою таких робіт.

Тема КП повинна бути актуальною, мати теоретичне і прикладне значення, відповідати сучасному стану проектування ПЗ, розробки інформаційних технологій та інженерії програмного забезпечення.

Тематика курсових проектів, рекомендована в межах спеціальності (див. Додаток Г), орієнтована на розв'язання складних інженерних задач із застосуванням сучасних методів візуального моделювання та спрямована на проектування і реалізацію:

1. Масштабованих архітектурних рішень: проектування розподілених, хмарних (Cloud-native) та мікросервісних архітектур програмного забезпечення для різних платформ із застосуванням сучасних патернів.
2. Реінжинірингу та модернізації ПЗ: аналіз існуючих складних систем, рефакторинг архітектури, зворотне проектування (Reverse Engineering) та міграція на нові технологічні стеки або платформи.
3. Інструментальних засобів підтримки життєвого циклу ПЗ: розробка спеціалізованого програмного забезпечення для автоматизації процесів розробки (CASE-засоби), розумного тестування (QA Automation), розгортання (CI/CD) та моніторингу систем.
4. Спеціалізованих програмних бібліотек та фреймворків: проектування базових програмних модулів для вирішення складних алгоритмічних або інфраструктурних задач під конкретні мови програмування чи середовища розробки.

5. Модулів розширення та інтеграційних шлюзів: розробка плагінів для існуючих корпоративних систем (ERP, CRM тощо), що суттєво розширюють їхній функціонал, з обов'язковим моделюванням процесів інтеграції (API, Webhooks).
6. Високонавантажених сервісів: проектування та розробка програмних сервісів для розв'язання наукоємних та спеціалізованих прикладних задач.
7. Програмного забезпечення з вираженою науково-інженерною складовою: розробка рішень, де архітектура та реалізація суттєво відрізняються від загальноприйнятих (типових) підходів. Це вимагає нестандартних архітектурних кроків для досягнення специфічних нефункціональних вимог (максимальна продуктивність, підвищена безпека, відмовостійкість), які неможливо забезпечити базовими рішеннями.

Організаційно процес виконання курсового проекту інтегровано з ітеративною моделлю розробки і складається з трьох ключових макроетапів, що частково відображають фази RUP:

- **Підготовчий** етап (Inception / початок Elaboration): аналіз предметної області, формування бачення (Vision) та бізнес-вимог, вибір базової архітектури та CASE-інструментарію.
- **Основний** етап (Elaboration / Construction): детальне об'єктно-орієнтоване проектування (розробка UML-діаграм), проектування бази даних, розробка компонентів системи та реалізація стратегії тестування.
- **Заключний** етап (Transition): формування повного пакету проєктних артефактів (пояснювальної записки), підготовка презентаційних матеріалів та публічний захист курсового проєкту.

Організаційно процес виконання КП складається з наступних етапів: підготовчий, основний, заключний.

3.1 Підготовчий етап

Під час підготовчого етапу (що концептуально відповідає фазі Inception методології RUP) здобувач попередньо обирає тему, узгоджує її з керівником та отримує від нього індивідуальне завдання щодо проблем, які необхідно вирішити. На цьому етапі здійснюється ознайомлення зі станом проблеми, збирання фактичних матеріалів, проведення глибокого бізнес-аналізу та виявлення потреб зацікавлених осіб (Stakeholders), вивчення специфіки предметної області, а також огляд актуальних науково-технічних досліджень.

Рекомендації для здобувача другого (магістерського) рівня на підготовчому етапі:

Вибір напряму та керівника: На початку семестру (відповідно до графіка навчального процесу, оприлюдненого на сайті ТНТУ) здобувач обирає тематику роботи та керівника зі складу викладачів випускової кафедри.

Керівником КП магістра може бути призначений науково-педагогічний працівник (НПП) кафедри, який має науковий ступінь (вчене звання), або висококваліфікований фахівець-практик ІТ-галузі за профілем спеціальності 121. Якщо тема курсового проєкту є масштабною (наприклад, проєктування хмарної або мікросервісної архітектури) і потребує залучення декількох здобувачів, може виконуватися комплексний курсовий проєкт [1]. У такому разі між магістрами чітко розподіляються проєктні ролі згідно з методологією RUP (наприклад: System Analyst, Software Architect, Component Engineer).

Формування індивідуального завдання (Vision): Спільно з керівником формується індивідуальне завдання, яке визначає межі системи (Scope). Це завдання охоплює: ознайомлення зі станом проблеми, збирання фактичних матеріалів, проведення системного аналізу вимог (функціональних та нефункціональних, зокрема за моделлю FURPS+), а також первинне формування документа "Бачення" (Vision Document) і концептуальної моделі бізнес-прецедентів (Business Use-Case Model).

Завдання на курсовий проєкт: Теми курсових проєктів затверджує керівник на початку семестру. Відповідно до теми студент разом із керівником складає завдання на курсову роботу. У завданні вказується: тема курсового проєкту; термін здавання завершеної роботи на кафедру; початкові дані до роботи; зміст роботи (перелік питань, які слід розробити). Підписують завдання керівник курсового проєкту і студент. Допускається виконувати завдання з обох боків аркуша білого паперу формату А4 . Форма завдання наведена в додатку В.

У п. 1 завдання вказують тему курсового проєкту, яка має бути сформульована чітко й лаконічно. Тема є індивідуальною для кожного студента, її визначають на кафедрі з урахуванням тематики наукових досліджень і договорів.

У п. 2 вказують термін, до якого студент повинен здати керівникові закінчену роботу на перевірку.

У п. 3 формулюють мету розробки, дають короткий опис прикладної проблеми і завдань, які потрібно вирішити в курсовій роботі. Вказують вид розробки (програмний проєкт, бібліотеку класів тощо), задають формат вхідних і вихідних даних програми.

У п. 4 зазначають основний перелік питань (розділів і підрозділів пояснювальної записки), які повинні бути розроблені при виконанні курсового проєкту.

У п. 5 вказують конкретні технічні вимоги, параметри програми, формат вхідних і вихідних файлів, основні пункти меню, тип та версію програмного середовища, у якому розробляють та виконують програму.

У п. 6 вказують дату затвердження завдання на курсовий проєкт.

3.2 Основний етап

Під час основного етапу (що концептуально охоплює фази Elaboration та Construction за методологією RUP) магістрант виконує детальне об'єктно-орієнтоване проектування, розробляє архітектурні моделі, реалізує базовий функціонал (Executable Architecture) у повному обсязі та направляє проміжні і фінальні артефакти на перевірку керівнику.

Обов'язки керівника КП:

- Фахове консультування: надає здобувачеві експертні роз'яснення щодо вибору архітектурних стилів, патернів проектування (GoF, GRASP, архітектурні патерни) та коректного застосування нотації UML 2.5 у вибраних CASE-засобах.
- Затвердження завдання: підписує здобувачеві індивідуальне завдання на КП у регламентовані терміни (додаток Б), визначаючи ключові ітерації розробки.
- Інформаційна підтримка: надає рекомендації щодо опрацювання актуальної літератури, міжнародних стандартів (IEEE, ISO/IEC), наукових публікацій та офіційних специфікацій з інженерії програмного забезпечення за обраною тематикою.
- Моніторинг прогресу: контролює реалізацію ітеративного плану виконання проєкту (Iteration Plan) та своєчасне виявлення і пом'якшення проєктних ризиків.
- Контроль дисципліни: у разі суттєвих порушень графіка, критичної невідповідності роботи магістерському рівню або виявлення академічного плагіату інформує завідувача кафедри для вжиття відповідних заходів, аж до недопущення КП до захисту.
- Науково-технічне керівництво: здійснює загальне кураторство процесом системного аналізу та архітектурного проектування.
- Архітектурні рев'ю: використовує відведений розкладом час для регулярних консультацій, під час яких здобувач звітує про стан виконання ітерацій. На цих зустрічах проводиться аудит UML-моделей, обговорюються альтернативні варіанти дизайну (Trade-off analysis) та здійснюється перевірка консистентності розроблених артефактів.
- Підготовка до захисту: консультує щодо структури презентації та правильного оформлення Документа архітектури програмного забезпечення (Software Architecture Document — SAD).

Основний етап (який плавно переходить у фазу Transition). На цьому етапі розробка артефактів проєкту (діаграм, специфікацій, прототипів) має бути

повністю завершена, оформлена у вигляді пояснювальної записки та офіційно погоджена керівником.

3.3 Заключний етап

Під час заключного етапу (що концептуально відповідає завершенню фази Transition за методологією RUP) відбувається фіналізація всіх проєктних артефактів, підготовка до публічної апробації та безпосередній захист результатів проєктування.

Основні заходи на заключному етапі:

1. Академічний контроль: обов'язкова перевірка фінальної версії пояснювальної записки (зокрема, Документа архітектури програмного забезпечення — SAD) на наявність академічного плагіату за допомогою спеціалізованих університетських систем.
2. Доопрацювання матеріалів: врахування зауважень керівника, усунення недоліків у UML-моделях, фіналізація архітектурних рішень та шліфування презентаційних матеріалів для захисту..

Науково-дослідна складова (для здобувачів рівня «Магістр»):

Оскільки курсові проєкти на другому освітньому рівні обов'язково містять елементи наукового дослідження (наприклад, порівняльний аналіз архітектурних стилів, дослідження ефективності патернів чи оптимізація алгоритмів інтеграції), вагомою складовою підготовки є наукова апробація. Здобувачам настійно рекомендується оприлюднення основних результатів КП у вигляді доповіді на науково-практичних конференціях із подальшою публікацією тез або фахової статті. Ці матеріали стануть суттєвим підґрунтям для майбутньої кваліфікаційної роботи магістра.

Завершення етапу:

Етап фіналізується проходженням нормоконтролю, друком та зшиванням паперової версії пояснювальної записки (разом із роздрукованим ілюстративним матеріалом та презентацією). Проєкт вважається повністю готовим до захисту після отримання всіх необхідних підписів (здобувача, керівника) на титульному аркуші та в завданні на курсовий проєкт. Також передбачається задача електронної версії артефактів (файли моделей з CASE-засобів, текст пояснювальної записки, презентація) до архіву кафедри.

3.5 Права та обов'язки здобувача

Здобувач зобов'язаний [3, 4, 5]:

Організація роботи: Своєчасно обрати керівника з числа науково-педагогічних працівників (НПП) кафедри програмної інженерії, узгодити з ним тему КП, що відповідає складності магістерського рівня, та отримати

індивідуальне завдання на проведення системного аналізу і збір артефактів предметної області.

Планування (Ітераційний підхід): Скласти та узгодити з керівником календарний (ітераційний) план виконання КП згідно з фазами методології RUP. Регулярно інформувати керівника про стан розробки архітектурних моделей на кожній ітерації та своєчасно надавати матеріали для перевірки (архітектурних рев'ю).

Самостійність: Самостійно виконувати проєкт або свою чітко визначену рольову частину (наприклад, Software Architect, System Analyst, Database Designer) у разі виконання комплексного КП.

Науково-технічний рівень: Під час розв'язання проєктних задач критично оцінювати та використовувати сучасні досягнення інженерії програмного забезпечення. Застосовувати передові архітектурні патерни, принципи об'єктно-орієнтованого аналізу та дизайну (OOAD), а також методологію RUP для прийняття обґрунтованих рішень щодо структури системи.

Використання CASE-засобів: При проєктуванні та генерації необхідних артефактів для різних рівнів моделювання використовувати професійні інструментальні засоби архітектурного проєктування (наприклад, Sparx Enterprise Architect, Visual Paradigm, IBM Engineering Systems Design Rhapsody, інструментарій екосистем JetBrains або Microsoft Visual Studio тощо).

Відповідальність за якість: Нести персональну відповідальність за правильність та життєздатність прийнятих архітектурних рішень, відсутність академічного плагіату та фабрикації даних. Забезпечити високу якість оформлення текстової частини (зокрема Software Architecture Document) та UML-діаграм, їх повну відповідність стандарту UML 2.5, методичним вказівкам кафедри програмної інженерії та чинним нормативним документам (ДСТУ/ISO/IEEE).

Дотримання регламенту: Строго дотримуватися графіка виконання КП, встановлених правил поведінки в лабораторіях та аудиторіях кафедри програмної інженерії, а також адекватно та конструктивно реагувати на зауваження керівника.

Подання роботи: У встановлений термін (до завершення фази Transition) подати всі артефакти КП керівнику для перевірки. Після усунення виявлених недоліків повернути роботу для остаточного схвалення та допуску до захисту.

Академічна доброчесність: Здійснити обов'язкову перевірку електронної версії пояснювальної записки на наявність текстових збігів (плагіату) у відповідній системі для підтвердження оригінальності дослідження.

Захист: Своєчасно прибути на публічний захист КП або завчасно попередити завідувача випускової кафедри та керівника про неможливість присутності з поважних причин.

Здобувач має право [3, 4, 5]:

- Вибір та адаптація теми: Обирати тему курсового проекту з актуального переліку, затвердженого кафедрою програмної інженерії, або пропонувати власну (зокрема ту, що безпосередньо пов'язана з його науковими дослідженнями чи професійною діяльністю в ІТ-галузі). При цьому необхідно обґрунтувати доцільність її розроблення та відповідність складності магістерському рівню. У разі об'єктивної необхідності здобувач може ініціювати питання щодо уточнення чи зміни теми КП у регламентовані навчальним планом терміни.
- Доступ до інфраструктури: Використовувати аудиторний фонд та спеціалізовані навчально-наукові лабораторії університету для роботи над проектом у відведений розкладом час.
- Використання ресурсів кафедри: Користуватися сучасною лабораторною та інформаційною базою, серверним і комп'ютерним обладнанням, а також ліцензійними CASE-засобами (інструментами для об'єктно-орієнтованого моделювання) та іншим спеціалізованим забезпеченням випускової кафедри для проведення досліджень, проектування архітектури та тестування розроблених моделей ПЗ.
- Фаховий супровід: Отримувати систематичні консультації керівника курсового проекту згідно із затвердженим розкладом для обговорення результатів кожної ітерації (за методологією RUP), аналізу архітектурних ризиків та верифікації проектних рішень.
- Архітектурна та технічна свобода: Самостійно обирати технологічний стек, архітектурні патерни та методи розв'язання поставлених інженерних задач, за умови їх належного науково-технічного обґрунтування та відповідності вимогам стійкого проектування програмних систем..

4 ВИМОГИ ДО ЗМІСТУ ТА СТРУКТУРИ КП

Пояснювальна записка (записка) повинна розкривати зміст курсового проекту, містити обґрунтування класифікації основних абстракцій та вибору архітектури розроблюваної системи, сценарії реалізації, методів, алгоритмів, засобів стандартної бібліотеки параметризованих класів тощо для виконання поставленого завдання, аналіз отриманих результатів та інші матеріали.

Матеріал пояснювальної записки повинен бути викладений грамотно, чітко та стисло. При цьому в тексті мають бути обов'язковими посилання на використані літературні та інші джерела.

У тексті пояснювальної записки не рекомендується вживати звороти із займенниками першої особи, наприклад: "Я вважаю ...", "Ми вважаємо ..." тощо. Рекомендується вести виклад, не вживаючи займенників, наприклад: "вважаємо

...", "... знаходимо ..." тощо. Без пояснень дозволяється використовувати тільки загальноприйняті скорочення, наприклад: ПК, ДСТУ, ООП тощо.

Пояснювальну записку поділяють на розділи і підрозділи, пункти і підпункти. Розділи в межах усієї пояснювальної записки повинні мати порядкові номери, позначені арабськими цифрами без крапки. Вступ не нумерують. Підрозділи повинні мати нумерацію в межах розділу: номер підрозділу складається з номера розділу і підрозділу, розділених крапкою, наприклад, 2.3 означає: третій підрозділ другого розділу. У кінці порядкового номера розділу, підрозділу тощо крапку не ставлять. Номер пункту містить номер розділу, підрозділу і пункту, які розділені крапками, наприклад, 3.2.1 – перший пункт другого підрозділу третього розділу. Для забезпечення стислості викладу та економії паперу назви пунктів / підпунктів можна не нумерувати, а вставляти їх в основний текст на початку абзацу, виділивши жирним шрифтом.

Назви розділів повинні бути короткими і записувати їх слід у вигляді заголовків прописними буквами посередині рядка. Назви підрозділів записують у вигляді заголовків рядковими буквами (перша прописна). Переноси слів у заголовках не допускаються. Крапку в кінці заголовка не ставлять. Між назвами розділів, підрозділів і основним текстом повинен бути пропущений рядок.

Заголовки розділів відділяють від тексту зверху й знизу трьома інтервалами (30..36 пунктів на комп'ютері). Абзаци в тексті починають відступом, що дорівнює п'яти пропускам. (13-17 мм).

Фрагменти коду, що ілюструють певні елементи програмної реалізації, оператори, назви класів, методів тощо, що подаються в тексті, доцільніше для зручності читання набирати шрифтом Arial.

Графічний ілюстративний матеріал у тексті ПЗ (схеми, ескізи, графіки, рисунки) виконують у графічному редакторі. Кількість ілюстрацій повинна бути достатньою для пояснення тексту, що викладається. Ілюстрації розміщують одразу після посилання на них за текстом ПЗ. Усі розміщені в ПЗ ілюстрації нумерують арабськими цифрами в межах одного розділу, наприклад, Рис. 2.3 - розділ 2, рисунок 3. Посилання на ілюстрації подають за типом: на рис. 2.3, повторно – див. рис. 1.3.

Помилки, описки і графічні неточності, виявлені в процесі виконання документа, допускається зафарбовувати коректором і нанесенням на тому ж місці виправленого тексту (графіки) машинописним способом чи чорною пастою рукописним способом.

Пошкодження аркушів пояснювальної записки, помарки і сліди неповністю видаленого попереднього тексту (графіки) не допускаються.

Порядкові числівники, які йдуть один за одним, можуть бути написані цифрами з відмінковим закінченням, яке ставлять лише при останній цифрі, наприклад: 1-е; 7, 8, 9-й тощо.

Перелік використаної літератури повинен містити лише ті літературні джерела, які використані при виконанні курсового проекту і на які є посилання в тексті документа.

Робота повинна бути написана державною мовою.

Зміст та структура КП визначається вимогами освітньо-професійної програми [9], завданням на КП та рекомендаціями Положення «Про підсумковий семестровий контроль результатів навчання студентів Тернопільського національного технічного університету імені Івана Пулюя» [1]. Структура, порядок слідування, найменування, об'єм структурних елементів та загальний об'єм КП, зміст типових розділів КП повинні відповідати вимогам, наведеним в таблиці 2.1 (рекомендації до розгорнутого змісту в Додатку Д).

Таблиця 2.1 Рекомендована структура та об'єм курсового проекту

Найменування структурного елемента роботи	Кількість сторінок	Перевірка на антиплагіат
Титульний аркуш	1	-
Завдання на КП (двохсторонній аркуш) (Додаток Б)	2	-
Анотація (додаток Д) сторінка №3	1	+
Abstract (англійською мовою)	1	+
Перелік умовних позначень, символів, одиниць, скорочень і термінів (за необхідності)	1-2	-
Зміст	1-2	
Вступ	1-2	+
Розділ 1 Аналіз вимог до програмної системи	5-10	+
Розділ 2 Проектування та архітектура програмної системи	10-15	+
Розділ 3 Розробка програмного коду та тестування системи	7-12	+
Висновки	1-2	+
Список використаних джерел (15- 20 джерел)	2-4	-
Додатки	-	-
Загальний обсяг пояснювальної записки (без додатків)	35-45 стор.	

Конкретні назви розділів, пунктів та підпунктів роботи формуються в залежності від теми КП. При цьому, КП повинен відповідати освітньо-

професійній програмі з описом основних етапів життєвого циклу програмного забезпечення. Слід також надати опис інтерфейсу взаємодії з користувачем.

Рекомендується виклад основного матеріалу роботи в трьох розділах. Рекомендований обсяг текстової частини КП складає 35-40 аркушів формату А4 без додатків. Одна сторінка орієнтовно 2000 символів.

Під час виконання основних розділів КП та оформлення супровідної документації рекомендується враховувати рекомендації та кращі практики інженерії програмного забезпечення [19, 20, 21, 22, 23]. Нормоконтроль курсового проєкту виконано відповідно до вимог ДСТУ 3008:2015. Архітектурні рішення документовано згідно з рекомендаціями стандарту ISO/IEC/IEEE 42010, моделювання програмної системи здійснено з використанням нотації UML (ISO/IEC 19505), а аналіз атрибутів якості проведено на базі моделі ISO/IEC 25010 [24, 25, 27, 28].

До комплексних курсових проєктів додатково висуваються наступні вимоги:

1. Комплексний КП оформляється у вигляді однієї пояснювальної записки, обсяг основної частини якої не менше як в 1,5 рази більший від обсягу основної частини звичайної КП. Зміст КП повинен бути логічно структуровано таким чином, щоб не тільки чітко подати суть і результати проведених досліджень, а й розкрити авторський внесок кожного з авторів по окремих пунктах та підпунктах;
2. На титульній сторінці КП необхідно вказати обох авторів;
3. В пояснювальній записці підшиваються два бланки "Завдання" (Окремий бланк для кожного з авторів). Бланки "Завдання" мають номери сторінок 2 та 3, що потрібно врахувати при загальній нумерації сторінок КП;
4. У вступі до КП обов'язково має бути висвітлено внесок кожного з авторів із зазначенням переліків авторства розділів та/або пунктів (підпунктів);
5. Список використаних джерел повинен включати не менше 15-20 найменувань (для комплексного КП в 1,5 рази більше від кількості джерел у звичайному КП).

5 ВИМОГИ ДО ОФОРМЛЕННЯ

5.1 Загальні вимоги

Назва теми КП має бути якомога точною, коротшою, лаконічною і відповідати змісту КП (до 12-и слів). Не можна вживати в назві теми КП невизначені формулювання, наприклад: «Аналіз деяких питань...», а також формулювань на кшталт: «До питання про...», «До вивчення...», «Матеріали до...».

КП повинна складатися з текстової частини (пояснювальної записки) та графічного (ілюстративного) матеріалу, який рекомендується оформляти у

вигляді слайдів презентації. Для обох цих документів мають надаватись примірником у PDF форматі, UTF-8 кодування, вбудовані шрифти, текстовий шар.

Пояснювальна записка до КП оформляється відповідно до вимог державних стандартів України:

- ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».
- ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».
- Рекомендується використовувати стиль оформлення списку наукових публікацій IEEE (Institute of Electrical and Electronics Engineers style) [18] як альтернативу ДСТУ 8302:2015

Пояснювальна записка оформляється на аркушах паперу білого кольору формату А4 з полями: 25 мм (з лівого боку), 10 мм (з правого боку), 20 мм (зверху і знизу) [1].

Параметри тексту: шрифт Times New Roman, чорного кольору, 14 кегль, міжрядковий інтервал 1,5, абзацний відступ 1,25 см., вирівнювання по ширині сторінки. Шрифт друку повинен бути чітким. Не можна використовувати згущений або розріджений шрифт.

Стиль викладення тексту роботи повинен бути науковим, чітким, без ліричних відступів, граматичних та орфографічних помилок.

Дозволяється в тексті ПЗ, крім заголовків, слова та словосполучення скорочувати згідно з правописними нормами та ДСТУ 3582:2013.

Обов'язкові структурні елементи **АНОТАЦІЯ, ABSTRACT, ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, ЗМІСТ, ВСТУП, СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ, ДОДАТКИ** не нумеруються. Аркуші слід нумерувати арабськими цифрами, додержуючись наскрізної нумерації впродовж усього тексту роботи. Номер аркуша проставляють у правому верхньому куті сторінки без крапки в кінці. При цьому на титульному аркуші та бланку ЗАВДАННЯ номер не проставляють, але їх включають до загальної нумерації аркушів КП. Бланк ЗАВДАННЯ вважається таким, що займає один аркуш. Ілюстрації і таблиці, розміщені на окремих сторінках, включають до загальної нумерації аркушів роботи.

В тексті КП слід уникати виділення фрагментів тексту жирним та/або курсивом, за винятком цитат (оформляються курсивом) та назв розділів/підрозділів/пунктів (оформляються жирним шрифтом). Також не допускається використання підкреслення. Цитату потрібно також вміщувати в подвійні лапки. Після цитати має бути посилання на джерело оформлене згідно стандарту ДСТУ 8302:2015 [15]. Посилання на літературу не повинно опинитися на початку нового рядка, тому слід використовувати нерозривний пробіл

(Ctrl+Alt+Space), щоб прив'язати дужки посилання до останнього слова в реченні. При посиланні на структурні елементи самої записки КП, зазначають відповідно номери розділів, підрозділів, пунктів, підпунктів, позицій переліків, рисунків, формул, рівнянь, таблиць, додатків. Посилаючись, треба використовувати такі вирази: «у розділі 4», «див. 2.1», «відповідно до 2.3.4.1», «(рисунок 1.3)», «відповідно до таблиці 3.2», «згідно з формулою (3.1)», «у рівняннях (1.23 – (1.25))», «(додаток Г)» тощо.

Нумерацію сторінок записки починають із титульного аркуша, на якому номер не проставляють. Аркуш, розміщений після завдання на КР, нумерують цифрою 3.

В загальний об'єм КП входять сторінки, починаючи від титульної і закінчуючи сторінками із списком використаних літературних джерел.

Текст основної частини роботи поділяють на розділи, підрозділи та пункти. Допускається, але не рекомендується використання в КП підпунктів (заголовків четвертого рівня).

Заголовки структурних елементів КП та заголовки розділів слід вирівнювати посередині рядка і друкувати **великими** літерами без крапки в кінці, не підкреслюючи. Заголовки розділів, підрозділів та пунктів повинні бути чіткими та лаконічними. Якщо заголовок складається з двох і більше речень, їх розділяють крапкою. Перенесення слів у заголовок не допускається. Не допускається розміщувати назву розділу, підрозділу чи пункту в нижній частині сторінки, якщо після неї розміщено тільки один рядок тексту. Розділи, підрозділи та пункти КП слід нумерувати арабськими цифрами. Для розділів і підрозділів ПЗ наявність заголовка обов'язкова. Пункти й підпункти можуть мати заголовки. Розділи роботи повинні мати порядкову нумерацію в межах викладення суті роботи і позначатись арабськими цифрами без Крапки, наприклад: **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Розділ це найбільша структурна частина роботи (наприклад, “1 АНАЛІЗ ВИМОГ В ПРЕДМЕТНІЙ ОБЛАСТІ”). **Підрозділ** це частина розділу (наприклад, “1.1 Огляд програмних рішень ...”). **Пункт** це деталізація всередині підрозділу (наприклад, “1.1.1 Основні визначення”). **Підпункт** це ще глибший рівень (наприклад, “1.1.1.1 Особливості застосування”).

Заголовки структурних елементів ПЗ та заголовки розділів потрібно друкувати посередині рядка великими літерами напівжирним шрифтом без Крапки в кінці. Заголовки підрозділів, пункти і підпункти друкувати з абзацного відступу з великої літери без крапки в кінці.

Після назви структурного елементу ПЗ, розділу, підрозділу, пункту обов'язковим є **порожній рядок** перед початком тексту. Після назви підпункту порожній рядок не ставиться. **Перед назвою** підрозділу, пункту, підпункту обов'язковим є порожній рядок крім випадків коли ці елементи опиняються на

початку сторінки (назви структурних елементів і розділів завжди на початку сторінки). Розділ обов'язково повинен мати текстовий вступ, хоча б кілька речень. Тобто не можна, щоб після назви розділу одразу йшла назва підрозділу без жодного тексту. Навіть якщо текст у вступі до розділу мінімальний (2–3 речення), він повинен бути. Це правило діє для всіх рівнів: після розділу та підрозділу має бути хоча б вступний текст, перш ніж починати пункти/підпункти.

Не дозволено розміщувати назву підрозділу, а також пункту й підпункту на останньому рядку сторінки.

Кожен розділ роботи необхідно починати з нової сторінки (гаряча комбінація для розриву сторінки Ctrl+Enter). Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу в межах розділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад: **1.1 Аналіз вимог до розробки ПЗ** Аналогічно, пункти повинні мати порядкову нумерацію в межах конкретного підрозділу, наприклад: **2.1.1 Розробка діаграми ієрархії класів архітектурної моделі системи** Заголовки підрозділів та пунктів друкують маленькими літерами (крім першої великої) жирним шрифтом з абзацного відступу. Розділ, підрозділ та пункт не повинен закінчуватись рисунком, таблицею, лістингом, маркованим або нумерованим списком, крім випадків, коли цей рисунок, таблиця, лістинг, маркований або нумерований список є прямим результатом даного розділу, підрозділу чи пункту.

В текстовій та графічній частинах роботи допускаються виправлення дрібних помилок та неточностей, акуратно виконані коректором та ручною (маркером, чорнилом) чорного кольору.

5.2 Вимоги до обов'язкових структурних елементів КП

Кожен обов'язковий структурний елемент КП необхідно починати з нової сторінки. На титульному аркуші (стандартний бланк, додаток А) повинні бути всі необхідні прізвища відповідальних осіб та їх підписи. Індивідуальне завдання на КП (стандартний бланк з календарним планом, **двосторонній** лист формату А4, Додаток Б) містить інформацію про тему роботи (в т.ч. дату затвердження теми, термін здачі студентом закінченої роботи, вихідні відомості до завдання; перелік питань, як необхідно висвітлити в роботі, перелік графічного матеріалу, перелік консультантів із розділів роботи, календарний план).

Завдання підписується керівником, здобувачем і затверджується завідувачем кафедри.

Сторінка з анотацією виконується українською та англійською мовами і повинна містити:

1. Загальну інформацію про роботу (П.І.П. здобувача, тема роботи, назва навчального закладу, факультет, кафедра, група, місто, кількість сторінок роботи, кількість рисунків, таблиць, слайдів графічного матеріалу, додатків, найменувань в списку використаних літературних джерел).
2. Перелік ключових слів.
3. Стисло характеристику змісту виконаних розділів роботи.

Сукупність ключових слів повинна відображувати основний зміст роботи. Загальна кількість ключових слів повинна бути не меншою трьох і не більшою двадцяти. Ключові слова подають у називному відмінку, друкують в рядок, через кому.

Характеристика змісту розділів роботи повинна бути стислою і точною. Необхідно використовувати синтаксичні конструкції, притаманні мові наукових та ділових документів, уникати складних граматичних зворотів. Також необхідно використовувати стандартизовану термінологію, уникати маловідомих термінів та символів.

Якщо в КП вживаються специфічні терміни, а також використано маловідомі скорочення, абревіатури, нові символи, позначення та ін., то їх перелік подається в роботі на окремій сторінці **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**, яку розміщують перед змістом.

Заголовок **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ** слід скоротити відповідно до вмісту. Перелік треба подавати в дві колонки, де зліва в алфавітному порядку наводять скорочення (термін, символ), справа – їх детальне пояснення. Спершу наводяться терміни українською мовою, після них – англійською. При цьому, якщо спеціальні терміни, скорочення, символи чи позначення повторюються в тексті роботи менше трьох разів, то їх в перелік не включають, а відповідне пояснення їх значення наводять у тексті КП відразу при першій згадці.

Зміст КП повинен містити найменування та номери початкових сторінок усіх розділів, підрозділів та пунктів, а також обов'язкових структурних елементів КП (від елемента ВСТУП до елемента СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ включно). Зміст ПЗ за ДСТУ 3008:2015 рекомендується оформляти з уступом номерів заголовків підрозділів на два знаки всередину тексту змісту.

Сторінка ВСТУП повинна містити такі відомості:

1. Коротке формулювання актуальності теми КП.
2. Сформульовані мета і перелік задач проектування/дослідження.
3. Сформульовані об'єкт і предмет дослідження.

4. Сформульовані новизна та практичне значення одержаних результатів (за наявності в КП).
5. Апробація курсового проєкту (за наявності в КП).
6. Короткі відомості про науково-технічні публікації (статті, тези, патенти чи авторські свідоцтва на твір, за умови їх наявності), підготовлені та опубліковані за результатами виконання КП.

Кожен розділ КП повинен закінчуватися абзацом з короткими висновками до нього.

На сторінці **ВИСНОВКИ** описують основні результати, одержані в КП, які повинні містити формулювання розв'язаної в роботі науково-технічної задачі, її теоретичне та практичне значення. У висновках необхідно наголосити на якісних і кількісних показниках отриманих результатів, обґрунтувати достовірність результатів, викласти рекомендації щодо їх практичного використання (за наявності можливості такого використання).

5.3 Оформлення списків

Загальні вимоги. Для всіх списків, позначених маркерами, цифрами (порядковими номерами), або комбінованих (багаторівневих списків, які включають як порядковий номер так і маркер) номер (маркер) першого рівня повинен знаходитися на відстані 1,25 см від лівого краю (абзацний відступ). Наприклад:

- Елемент списку з маркером.
- Елемент списку з маркером.

Елемент списку повинен починатись з великої букви та завершуватись крапкою. Не допускається починати частину елементів одного й того ж списку з великої букви, а іншу частину – з малої.

Якщо елемент списку містить одне речення, або слово – він може починатись з маленької букви та завершуватись крапкою з комою. Наприклад:

- елемент списку з маркером, з маленької букви;
- елемент списку з маркером, з маленької букви;
- останній елемент такого списку повинен завершуватись крапкою.

або

- 1) елемент нумерованого списку з маленької букви;
- 2) елемент нумерованого списку з маленької букви;
- 3) останній елемент такого списку повинен завершуватись крапкою.

Не рекомендується використання багаторівневих списків із кількістю рівнів більше двох, в тому числі багаторівневих комбінованих списків.

Для багаторівневих маркованих списків слід використовувати такий самий маркер, як для і маркованих. Стиль багаторівневих нумерованих списків повинен бути подібним на стиль однорівневих нумерованих. Марковані списки.

В якості маркера можна використовувати тире “–” або велику крапку “●”. Не допускається використовувати частину елементів одного й того ж списку позначених тире, а іншу частину – крапками. В межах всієї пояснювальної записки потрібно використовувати один і той же маркер для маркованих списків. Нумеровані списки. Елементи таких списків позначаються арабськими цифрами. Нумерований список слід форматувати таким чином:

1. Елемент нумерованого списку.
2. Елемент нумерованого списку.
3. Елемент нумерованого списку.

Другий спосіб форматування нумерованого списку включає використання порядкового номера з дужкою:

- 1) елемент нумерованого списку з маленької букви;
- 2) елемент нумерованого списку з маленької букви;
- 3) останній елемент такого списку повинен завершуватись крапкою.

При цьому кожен елемент такого списку повинен починатися з малої букви та завершуватися крапкою з комою окрім останнього елемента, який повинен завершуватися крапкою. В межах всієї пояснювальної записки потрібно використовувати один і той же стиль оформлення нумерованих списків.

5.4 Оформлення ілюстрацій

Ілюстрації (рисунок, графіки, схеми, діаграми, фотографії тощо) слід розміщувати в курсовому проєкті безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці (якщо ілюстрація не вміщається після згадки про неї на поточній сторінці). Якщо рисунки (ілюстрації) створені не автором КП, то подаючи їх у ПЗ, потрібно посилатись на них, дотримуватись чинного законодавства України про авторське право.

Ілюстрації повинні мати назву, яку розміщують одразу під ілюстрацією. Підпис повинен бути на тій же сторінці що і рисунок. Назва рисунка має відображати його зміст, бути конкретною та стислою. За потреби під ілюстрацією розміщують пояснювальні дані (підрисунковий текст). Всі рисунки, в тому числі розміщені в додатках до роботи, повинні бути чіткими і читабельними. Надписи, присутні на рисунках, повинні бути приблизно того ж розміру (кегель), що й розмір шрифту основного тексту роботи.

Ілюстрація позначається словом «Рисунок», яке разом із назвою ілюстрації розміщують після пояснювальних даних, наприклад, «Рисунок 3.1 – Діаграма варіантів використання системи». Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком ілюстрацій, поданих у додатках. Номер ілюстрації складається з номера розділу та порядкового номера ілюстрації, відокремлених крапкою, наприклад, «Рисунок

3.2 – XXXXXXXXX» – другий рисунок третього розділу, або «Рисунок А.1 – XXXXXXXXX» – перший рисунок додатку А.

Перед рисунком та після підпису до нього слід залишити **один вільний рядок**. Як рисунок, так і його назва вирівнюються по середині без абзацного відступу.

На всі ілюстрації, в тому числі на розміщені в додатках, повинні бути посилання в тексті КП. Друге та наступні посилання в тексті на той же самий рисунок пишуться скорочено – рис.

5.5 Оформлення таблиць

Структурований цифровий або текстовий матеріал, як правило, оформлюють у вигляді таблиць. Таблицю необхідно розташовувати безпосередньо після тексту, у якому вона згадується вперше, або на наступній сторінці. Назва таблиці складається зі слова «Таблиця», її порядкового номера та безпосередньо назви, яка стисло характеризує зміст поданих у ній даних. Повну назву таблиці вказують один раз над таблицею зліва з абзацним відступом. На всі таблиці, в тому числі на розміщені в додатках, повинні бути посилання в тексті роботи (наприклад, “в таблиці 1.2”). Приклад другого, або наступного посилання на таблицю: в табл. 1.1.

Таблиці нумеруються арабськими цифрами підряд у межах розділу, за винятком таблиць, що подаються в додатках. Номер таблиці складається з номера розділу та порядкового номера таблиці в межах цього розділу, відокремлених крапкою, наприклад, «Таблиця 2.1 – XXXXXXXXXX» – перша таблиця другого розділу, «Таблиця В.2 – XXXXXXXXXX» – друга таблиця додатку В. Заголовки та дані в таблиці можуть бути виконані через один інтервал, шрифтом Times New Roman, 12 кегль. Заголовки колонок та/або рядків таблиці наводять жирним шрифтом, з великої літери, а підзаголовки – з малої, якщо вони становлять одне речення з заголовком. Підзаголовки, що мають самостійне значення, пишуть з великої літери. В кінці заголовків і підзаголовків таблиць крапки не ставлять. Заголовки та підзаголовки подають в однині. Приклад оформлення таблиці:

Таблиця 5.1 – Найменування таблиці

Заголовок 1	Заголовок 2	Заголовок 3	Заголовок 4
Текст	Текст	Текст	Текст

Перед підписом таблиці та після таблиці слід залишити **один вільний рядок**.

Рекомендується розміщувати таблицю повністю на сторінці, без переносу на наступну сторінку, вирівнюючи її по центру. Але, якщо таблиця велика і

переносу не наступну сторінку не уникнути, то перед перенесеною частиною таблиці з абзацного відступу пишуть: «Продовження таблиці Х.Х» або «Кінець таблиці Х.Х», де Х – номер таблиці. При перенесенні таблиці номери стовпців позначають латинськими цифрами і в перенесеній частині таблиці вказують саме номери стовпців, а не їх назви. Приклад оформлення перенесеної на наступну сторінку таблиці:

Продовження таблиці 5.1

1	2	3	4
Текст	Текст	Текст	Текст

5.6 Оформлення формул та рівнянь

Формули та рівняння розташовують безпосередньо після тексту, в якому вони згадуються вперше, посередині сторінки. Потрібно використовувати єдиний стиль для всіх формул та рівнянь в тексті пояснювальної записки (шрифт, розміри), при цьому формули та рівняння повинні бути чіткими та читабельними. Вище та нижче кожної формули (рівняння) потрібно залишити один вільний рядок.

Формули та рівняння в КП (за винятком формул і рівнянь, поданих у додатках) слід нумерувати порядковою нумерацією в межах розділу. Номер формули (рівняння) повинен складатися з номера розділу та порядкового номера формули (рівняння) в межах цього розділу, відокремлених крапкою. Номер формули (рівняння) зазначають на рівні самої формули (рівняння) в дужках у крайньому правому положенні рядка.

У формулах і/або рівняннях верхні та нижні індекси, а також показники степеню, в усьому тексті ПЗ мають бути однакового розміру, але меншими за букву чи символ, якого вони стосуються.

Пояснення значень символів і числових коефіцієнтів, що входять до формули (рівняння), слід наводити безпосередньо під формулою (рівнянням) в тій послідовності, у якій вони наведені в ній. Пояснення значення кожного символу та числового коефіцієнта слід наводити з нового рядка. Перший рядок пояснення починають словом «де» без двокрапки і без абзацу. Безпосередньо після формули (рівняння) слід ставити кому (якщо за нею слідує ще одна формула/рівняння, або пояснення символів/числових коефіцієнтів, що входять до неї) або крапку (якщо за нею слідує новий абзац тексту). Приклад оформлення формули:

$$S = \frac{1}{p + \frac{1-p}{n}}, \quad (5.1)$$

де S — показник швидкодії виконання програми (як відношення до її початкового часу роботи), p — частина яку можна виконувати послідовно, $1 - p$ — частина, яка виконується паралельно, n — кількість процесорів.

Переносити формули (рівняння) на наступний рядок допускається тільки на знаках виконуваних операцій, повторюючи знак операції.

Допускається написання загальновідомих формул (рівнянь) безпосередньо в межах тексту роботи у випадку, коли на ці формули немає посилань в тексті.

Числові значення величин з допусками наводять так:

$$(65 \pm 3)\%; 80 \text{ м} \pm 2 \text{ м} \text{ або } (80 \pm 2) \text{ м};$$

діапазон чисел фізичних величин наводять, використовуючи прикметники «від» і «до», наприклад: «Від 1 с до 5 с (а не від 1 до 5 с)»;

якщо потрібно зазначити два чи три виміри, їх подають так:

$$80 \text{ м} \times 25 \text{ м} \times 50 \text{ м} \text{ (а не } 80 \times 25 \times 50 \text{ м)}.$$

5.7 Оформлення лістингів програмного коду

Якщо програмний код займає три рядки або менше, то він не позначається заголовком “Лістинг” та не впливає на нумерацію лістингів. Код лістингу (як нумерованого, так і не нумерованого) форматується шрифтом Courier 12pt, одинарний міжрядковий інтервал, перший рядок коду без абзацного відступу. Наприклад:

Перший рядок програмного коду.

Другий рядок програмного коду

Третій рядок програмного коду

Якщо програмний код займає чотири рядки або більше, то він позначається заголовком Лістинг та номером в межах поточного розділу роботи. Наприклад:

Лістинг 1.1 – Програмний код

Перший рядок програмного коду.

Другий рядок програмного коду

Третій рядок програмного коду

Програмний код – наступний рядок

Програмний код – наступний рядок

В тексті КП обов’язково повинні бути посилання на всі лістинги програмного коду. Приклад звертання до лістингу в тексті: див. лістинг 1.1. Програмний код лістингу можна переносити на наступну сторінку без будь-яких позначень. При розірванні великого лістингу на дві або більше сторінок заголовок лістингу на наступній сторінці не повторюється. Перед

заголовком та після нумерованого лістингу слід залишити один вільний рядок. Перед та після не нумерованого лістингу, а також перед підписом і після нумерованого лістингу слід залишити один вільний **рядок**. Лістинг коду більше двадцяти рядків рекомендовано оформляти в додатках.

5.8 Оформлення посилань на літературні джерела

Посилання в тексті роботи на літературні джерела слід позначати порядковим номером, виділеним двома квадратними дужками, наприклад, «... у роботах [1], або [2,7], або [3-6]...». В курсовому проєкті обов'язково повинні бути посилання на всі без винятку наведені в списку літературні джерела. Список використаних літературних джерел в пояснювальній записці розміщується після загальних висновків по роботі (після сторінки ВИСНОВКИ). Список формується здобувачем за його вибором одним із таких способів:

- в порядку появи посилань у тексті;
- в алфавітному порядку прізвищ перших авторів або заголовків;
- у хронологічному порядку.

Бібліографічний опис списку використаних джерел у КП може оформлятися здобувачем за його вибором з урахуванням стилю оформлення списку наукових публікацій IEEE (Institute of Electrical and Electronics Engineers style) [18] або Національного стандарту України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» **Всі джерела повинні бути оформлені в одному стилі.**

Таблиця 5.2 – Зразки оформлення посилань на книги, однотомні видання згідно ДСТУ 8302:2015

Один автор	Мартін Р. Чиста архітектура. Мистецтво розробки програмного забезпечення./ Пер.з англ. І. Боднр-Терещенко. Харків, 2018. 368 с.
Два і більше авторів	Odersky M., Spoon L., Venners B. Programming in Scala. 3rd edition. Walnut Creek : Artima Press, 2017. 837 p. Lupenko S., Petryk M., Legrand A.P., Khimich O. High-performance technologies of modeling and identification of complex multi-component systems and processes. Studies and Monographs. z.594. Scientific Publishing of University Politechnika Opolska. Opole, Poland, 2024. 202 p.

Таблиця 5.3 – Зразки оформлення посилань на книги, багатотомні видання

Видання загалом	Енциклопедія історії України: у 10 т. Київ: Наук. думка, 2013. Т. 10. 784 с.
Окремий том	Dark energy and dark matter in the Universe. In three vol. Vol. 3. Dark matter: Observational manifestation and experimental searches. Kyiv: Akadempriodyka, 2015. 356 p.

Таблиця 5.4 – Зразки оформлення посилань на інші видання

Законодавчий матеріал	Про видавничу справу: Закон України від 05.06.1997 № 318/97-ВР. Київ: Парламентське видавництво, 2015. 24 с.
ЧАСТИНА ВИДАННЯ Стаття із журналу, збірника, розділ книги	
Незалежно від кількості авторів	Петрик М.Р., Бачинський М.В., Хіміч О.М., Бішак Д.С. Легранд А.-П. Інформаційна система цифрового аналізу мультисенсорних когнітивних впливів EEG-сигналів у випадку неврологічних розладів організму людини. Кібернетика та системний аналіз, 2026, том 62, № 1, С.122-136.
ЕЛЕКТРОННІ РЕСУРСИ ВІДДАЛЕНОГО ДОСТУПУ	
Опис ресурсу загалом	Наукові публікації і видавнича діяльність НАН України. Київ, 2007. URL: http://www.nas.gov.ua/publications (дата звернення: 19.03.2023).
Опис частини електронного ресурсу у позатекстовому списку бібліографічних посилань (списку літератури)	Garfield E. More on the ethics of scientific publication: abuses of authorship attribution and citation amnesia undermine the reward system of science. Essays of an information scientist. URL: http://www.garfield.library.upenn.edu/essays/v5p621y1981-82.pdf (Last accessed: 16.04.2024).

Список використаних літературних джерел до КП повинен містити 15-20 джерел (з яких 25% наукові публікації). При опрацюванні літератури потрібно щоб література відповідала тематиці КП. Звертайте увагу на актуальність інформації та віддавати перевагу сучасним джерелам — літературі виданій за останні 5-10 років. Забороняється використовувати джерела інформації створені на території держави-агресора, російськомовні джерела, та літературу за авторством росіян і білорусів [15].

5.9 Оформлення додатків

Фрагменти інформації, розміщення яких в основному тексті роботи є недоцільним (великі та об'ємні лістинги програмного коду, ілюстрації, рисунки, схеми, таблиці, документи, які займають декілька сторінок, протоколи проведених випробувань, копії опублікованих під час роботи над КП тез/статей конференцій, копії отриманих патентів/свідоцтв про авторське право на твір тощо), повинні бути оформлені у вигляді додатків. В основному тексті роботи потрібно обов'язково подати посилання на ці додатки, наприклад: «в додатку А», або ж посилання може бути подане в круглих дужках, наприклад «(див. додаток В)».

Додатки в роботі розміщують одразу після списку використаних джерел. Для цього на Окремому аркуші друкують великими літерами слово ДОДАТКИ, після чого на наступних аркушах розміщують необхідні додатки, кожен додаток з нової сторінки. Сторінки з додатками не нумерують і вони не враховуються при підрахунку загального об'єму КП. Додатки необхідно позначати послідовно великими літерами української абетки, за винятком літер Г, І, Є, І, Й, О, Ч, Ь, наприклад: додаток А, додаток Б і т.д. Один додаток позначається Окремою літерою. З правого боку рядка над заголовком малими літерами з першої великої друкується слово «Додаток __» і велика літера, що позначає додаток. Текст кожного додатка за необхідності може бути поділений на розділи і підрозділи, які нумерують у межах кожного додатка. Тоді перед кожним номером ставлять позначення додатка (літеру) і крапку, наприклад, А.2 – другий розділ Додатку А. Ілюстрації, таблиці і формули, розміщені в додатках, нумерують у межах кожного додатка, наприклад: «Рисунок Б.3» – третій рисунок додатка Б, або «Рисунок Д.1.2» – другий рисунок першого розділу додатка Д; «формула (В.1)» – перша формула додатка В.

5.10 Оформлення коду та супровідної документації

Код проєкту та супровідну документацію рекомендовано оформляти у вигляді GИT репозиторію та розмістити його в GitHub організації випускової кафедри (<https://github.com/orgs/TNTU-121-Software-Engineering>). Приклад назви репозиторію 2024-2025-KRB-SE-41-Ivan-PULUJ (**рік, ПІБ та групу змінити на свої дані**). Доступ до репозиторію має бути приватним та по завершенню захисту репозиторій потрібно заархівувати на GitHub.

Матеріали електронних версій курсового проєкту завантажуються у відповідну скриньку для завдань на файлах курсу [17].

Рекомендовано використовувати один репозиторій для КП.

6 ПОРЯДОК ПЕРЕВІРКИ ТА ЗАХИСТУ

6.1 Підготовка до захисту

Виконаний згідно з чинними інженерними стандартами, у повному обсязі та відповідно до індивідуального завдання курсовий проєкт (надалі – КП), підписаний здобувачем вищої освіти, подається на первинну перевірку керівникові у вигляді електронної версії пояснювальної записки та незброшурованої паперової чернетки (за вимогою).

Вимоги до електронних артефактів проєкту. Оскільки дисципліна базується на методології раціонального уніфікованого процесу (RUP), до пояснювальної записки обов'язково додається структурований електронний

архів (або посилання на репозиторій, наприклад GitHub/GitLab), який повинен містити:

- Файли архітектурних моделей: вихідні файли проєктів із використаних CASE-засобів (наприклад, .ear для Enterprise Architect, .vpp для Visual Paradigm тощо) з усіма розробленими UML-діаграмами.
- Документацію: електронну версію Документа архітектури програмного забезпечення (Software Architecture Document – SAD) та інші артефакти фаз Inception та Elaboration.
- Програмну реалізацію: вихідні коди розробленого архітектурного прототипу (Executable Architecture), згенерований CASE-засобом каркас коду або повноцінні модулі системи.
- Інфраструктурний код: скрипти створення бази даних (DDL), конфігураційні файли для розгортання (наприклад, Dockerfile, docker-compose.yml) та тестові специфікації.

Примітка: Якщо розроблений прототип вимагає специфічних бібліотек, фреймворків або системних залежностей, замість додавання бінарних файлів (DLL тощо) електронний архів повинен містити інструкцію з розгортання (файл README.md) із зазначенням команд для ініціалізації середовища та завантаження залежностей (через пакетні менеджери npm, pip, NuGet, Maven тощо).

Передача електронних матеріалів здійснюється шляхом завантаження в систему електронного навчання університету або надання доступу до хмарного сховища. У кореновому каталозі архіву необхідно розмістити інформаційний файл із зазначенням: прізвища та ініціалів магістранта, шифру академічної групи, дати подання та повної теми курсового проєкту.

Роботу необхідно подати на фінальну перевірку керівнику не пізніше, ніж за тиждень до захисту на кафедрі. Виявлені під час архітектурного ревію неточності, логічні помилки в UML-моделях чи дефекти оформлення здобувач зобов'язаний виправити у визначені керівником терміни. Якщо встановлено, що проєкт потребує суттєвого доопрацювання (наприклад, виявлено критичну невідповідність архітектури нефункціональним вимогам), визначається обсяг необхідних змін та встановлюється новий дедлайн.

До захисту категорично не допускаються курсові проєкти:

- що не відповідають затвердженій темі або рівню складності магістерської підготовки;
- подані без затвердженого завдання на КП, підписаного керівником та здобувачем;
- у яких за результатами автоматизованої або експертної перевірки виявлено академічний плагіат (запозичення з інших джерел без

належних посилань), фабрикацію або фальсифікацію результатів проєктування.

6.2 Академічна доброчесність

Учасники освітнього процесу під час провадження наукової діяльності навчання, викладання дотримуються та принципів академічної доброчесності [7] і усвідомлюють наслідки порушення цих принципів.

З метою підвищення якості навчання, розвитку навичок коректної роботи із джерелами інформації та формування звички до сумлінного дотримання вимог наукової етики, активізації самостійності та індивідуальності при створенні авторського твору та підвищення відповідальності здобувачів усіх форм навчання за порушення правил академічної етики діє Положення про недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя [6].

Дотримання академічної доброчесності здобувачами вищої освіти передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей);
- посилання на джерела інформації в разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної діяльності, використані методики досліджень і джерела інформації.

Порушенням академічної доброчесності учасниками освітнього процесу є:

- 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 використовують в освітньому процесі або наукових дослідженнях;
- фальсифікація – свідомо зміна чи модифікація вже наявних даних, що стосуються освітнього процесу чи наукових досліджень;
- списування – виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання, зокрема під час оцінювання результатів навчання здобувачів вищої освіти;
- обман – надання завідомо неправдивої інформації щодо власної освітньої (наукової, творчої) діяльності чи організації освітнього процесу;

– неправомірна вигода – грошові кошти або інше майно, переваги, пільги, послуги матеріального або нематеріального характеру, що їх обіцяють, пропонують, надають або одержують безоплатно чи за ціною, нижчою за мінімальну ринкову, без законних на те підстав, що призводить до необ'єктивності оцінювання;

– хабарництво – надання (отримання) учасником освітнього процесу чи пропозиція щодо надання (отримання) коштів, майна, послуг, пільг чи будьяких інших благ матеріального або нематеріального характеру з метою отримання неправомірної переваги в освітньому процесі, яке призводить на необ'єктивності оцінювання;

– необ'єктивне оцінювання – свідоме завищення або заниження оцінки результатів навчання здобувачів вищої освіти.

Дотримання принципів академічної доброчесності здобувачами вищої освіти регулюється Положенням про академічну доброчесність в Тернопільському національному технічному університеті імені Івана Пулюя, затверджене Вченою радою Університету (Наказ № 4/7-969 від 01.11.2019 р.)

6.2 Підготовка презентації та доповіді

Презентація курсового проєкту повинна бути змістовною та повністю розкривати суть виконаного завдання. Інформація, наведена на слайдах, не повинна повторювати текст доповіді, а лише доповнювати її та ілюструвати основні положення роботи та результати проведеного дослідження. На першому слайді презентації повинна бути вказана тема КП, прізвище та ім'я автора курсового проєкту, прізвище та ініціали керівника, його науковий ступінь і вчене звання. На наступних слайдах презентації рекомендовано чітко та лаконічно сформулювати та навести:

- актуальність тематики та короткий аналіз стану проблеми, що розглядається в КП;
- постановку задачі, яка вирішується в даній роботі (мета та задачі КП);
- специфікацію вимог до програмного забезпечення;
- UML-діаграму варіантів використання з їх текстовими описами;
- діаграма ієрархії класів предметної області
- Діаграма ієрархії класів / підсистем архітектурної моделі ПЗ;
- діаграми окремих класів для їх детального проєктування,
- діаграми послідовностей, що відображають сценарії реалізації основних варіантів використання ПЗ;
- діаграми діяльності, що відображають основну бізнес-логіку ПЗ та алгоритми реалізації Окремих методів;
- діаграми станів об'єктів класів керування (при необхідності);
- результати функціонального тестування програмного забезпечення;

- використані інструментальні засоби розробки ПЗ.

На передостанніх слайдах презентації наводяться:

- отримані практичні результати, інноваційність запропонованих рішень; публікації за темою КП, дані про використання (при наявності);
- висновки по КП.

Текст та ілюстрації чіткі і контрастні для оптимального відображення.

Нумерація слайдів внизу справа є обов'язковою, перший слайд без номера сторінки. Оформлення усіх слайдів узгодити в одній кольоровій схемі.

Презентація та доповідь повинні бути чітко узгоджені в часі. В процесі підготовки до захисту та виступу необхідно принаймні один раз проговорити доповідь вголос, одночасно запустивши презентацію, оцінити (а при потребі скоригувати) час виступу, який не повинен перевищувати 5 – 7 хвилин (для комплексних КП може бути збільшений до 10-12 хвилин)

Після доповіді здобувач вищої освіти стисло відповідає на запитання членів комісії. Необхідно зауважити, що у випадку комплексних курсових проєктів кожен з авторів роботи повинен бути обізнаний з загальною частиною роботи й готовий відповідати на питання комісії щодо неї.

Готові до передачі в архів файли передавати у спосіб визначений на попередньому захисті та розісланий на університетські скриньки здобувачів. Приклади назв файлів та каталогів подані у табл. 6.1

Таблиця 6.1 – Зразки оформлення назв файлів

Опис файлу/каталогу	Шлях та назва файлу/каталогу
Каталог який містить усі файли студента (рік, ПІБ та групу змінити на свої дані у всіх подальших файлах)	2025_KR_RUP_SE_51_I_PULUJ
Пояснювальна записка до КП зі всіма сторінками повністю (включно з заповненою титулкою), одним файлом у форматі .pdf з текстовим шаром, вбудованими шрифтами та кодування UTF-8. Файл готовий для внесення у архів кафедри.	/KK_dyplom_Puluj_I_2025.pdf
Презентація роботи у форматі .pdf з текстовим шаром, вбудованими шрифтами та кодування UTF-8.	2025_KP_RUP-SE-5-Іван-Павлович-ПУЛЮЙ-презентація.pdf
Пояснювальна записка до КП. Формат .pdf з текстовим шаром, вбудованими шрифтами та кодування UTF-8 для перевірки на антиплагіат, скорочена..	2025_KP_RUP-SE-5-Іван-Павлович-ПУЛЮЙ-антиплагіат.pdf
Програма, що виконується (усі файли із модулями, що виконуються, а також усі	2025_KR_RUP_SE_51_I_PULUJ - project-src.zip

необхідні бібліотеки якщо вони не публічно доступні). Каталог заархівований у форматі .zip без захисту паролем. Рекомендовано GIT репозиторій.	
Усі додаткові файли, котрі не були включені раніше варто додати до архіву у форматі .zip без захисту паролем.	2025_KR_RUP_SE_51_I_PULUJ - project-docs.zip

6.3 Порядок захисту курсового проєкту

До захисту курсового проєкту допускаються здобувачі вищої освіти, які в повному обсязі виконали вимоги навчальної програми та ітераційного календарного плану, своєчасно представили пояснювальну записку, архітектурні моделі та всі необхідні супровідні матеріали.

Захист курсових проєктів проводиться відкрито у формі публічної презентації. На захист роботу слід представляти у зброшурованому вигляді (паперовий примірник). Електронний варіант проєкту (згідно з вимогами розділу «Підготовка до захисту») має бути завантажений у систему дистанційного навчання університету або розміщений у погодженому репозиторії.

Процедура захисту курсового проєкту:

- Доповідь здобувача про ключові результати проєктування та архітектурні рішення (5–7 хвилин).
- Демонстрація артефактів: показ UML-моделей у CASE-засобі та демонстрація працездатності архітектурного прототипу (Executable Architecture).
- Відповіді здобувача на запитання членів комісії та присутніх.
- Обговорення результатів та оголошення оцінки.

Структура доповіді:

- Вступ: обґрунтування актуальності теми, стислий аналіз предметної області, визначення меж системи (Scope) та ключових задач проєктування.
- Основна частина: опис обраного архітектурного стилю та патернів, демонстрація основних представлень моделі «4+1» (Use Case View, Logical View, Deployment View тощо), обґрунтування вибору технологічного стеку та засобів автоматизації. Обов'язково наголошується на тому, як обрані рішення нівелюють проєктні ризики.
- Висновки: узагальнення результатів, підтвердження повноти реалізації вимог та відповідності розробленої архітектури критеріям якості (FURPS+).

При визначенні оцінки береться до уваги рівень теоретичної підготовки магістра, якість побудованих моделей, глибина архітектурного аналізу та здатність аргументовано захищати прийняті рішення.:

6.4 Оцінювання курсових проєктів

Оцінювання проєкту здійснюється членами комісії. При цьому приймаються до уваги:

- зміст роботи;
- якість оформлення;
- доповідь та презентація;
- відповіді на запитання членів комісії і присутніх;

Критерії оцінювання за національною шкалою та ECTS

Оцінка «Відмінно» (90–100 балів; A) виставляється, якщо:

- проєкт виконано на високому професійному рівні, суттєві недоліки відсутні.
- розроблена архітектура повністю відповідає функціональним та нефункціональним вимогам.
- здобувач продемонстрував глибоке володіння методологією RUP та об'єктно-орієнтованим аналізом і дизайном (OOAD).
- UML-моделі побудовані коректно, консистентно та відображають різні аспекти системи.
- здобувач впевнено та аргументовано відповідає на запитання, вільно володіє професійною термінологією та CASE-засобами.

Оцінка «Добре» (75–89 балів; B, C) виставляється, якщо:

- пояснювальна записка та моделі оформлені належним чином, проте є незначні зауваження до графічного матеріалу.
- архітектурні рішення є обґрунтованими, але прототип має певні обмеження у функціональності.
- у моделюванні використано основні принципи RUP, проте не всі переваги методології чи архітектурних патернів реалізовано в повному обсязі.
- під час захисту здобувач допустив незначні неточності у відповідях або недостатньо повно обґрунтував вибір окремих компонентів.

Оцінка «Задовільно» (60–74 бали; D, E) виставляється, якщо:

- робота містить суттєві недоліки в оформленні або структурі артефактів.
- проєкт лише частково відповідає вимогам, архітектура системи є надто спрощеною або неоптимальною для обраної теми.
- технологія RUP використана формально, відсутня чітка простежуваність (traceability) від вимог до моделей.

- здобувач відчуває труднощі під час захисту, не може повноцінно пояснити взаємодію елементів системи на діаграмах.

Оцінка «Незадовільно» (0–59 балів; FX, F) виставляється, якщо:

- проєкт не відповідає поставленому завданню або тема не розкрита.
- виявлено академічний плагіат або роботу виконано не самостійно.
- програмний прототип непрацездатний, а ключові UML-діаграми відсутні або містять критичні логічні помилки.
- здобувач не володіє матеріалом власного проєкту та не може відповісти на базові запитання комісії.

Результати захисту КП оцінюються за 100-бальною шкалою. Відповідність результатів, оцінених за 100-бальною шкалою національній шкалі («відмінно», «добре», «задовільно», «незадовільно») та шкалі Європейської кредитної трансферно-накопичувальної системи («A», «B», «C», «D», «E», «FX», «F») регламентується Положенням про оцінювання здобувачів вищої освіти в Тернопільському національному технічному університеті імені Івана Пулюя [10].

СПИСОК ДЖЕРЕЛ

1. Положення про кваліфікаційні роботи студентів Тернопільського національного технічного університету імені Івана Пулюя - наказ №4/7-241 від 15.04.2020 - зі змінами від 26.01.2021 - наказ №4-7-73 від 02.02.2021
URL: <https://docs.tntu.edu.ua/base/document?id=496>
2. Закон України “Про освіту”
URL: <https://zakon.rada.gov.ua/laws/show/2145-19#Text>
3. Закон України Про вищу освіту
URL: <https://zakon.rada.gov.ua/laws/show/1556-18#Text>
4. Положення «Про підсумковий семестровий контроль результатів навчання студентів Тернопільського національного технічного університету імені Івана Пулюя» - наказ №4/7-122 від 17.02.2020
URL: <https://docs.tntu.edu.ua/base/document?id=489>
5. Положення про організацію освітнього процесу в Тернопільському національному технічному університеті імені Івана Пулюя - наказ №4/7-651 від 21.06.2024
URL: <https://docs.tntu.edu.ua/base/document?id=1114>
6. Положення про недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя - наказ №4/7-964 від 01.11.2019 зі змінами від 19.12.2019 наказ №4/7-114 від 12.02.2020, зі змінами від 26.01.2021 - наказ №4/7-72 від 02.02.2021
URL: <https://docs.tntu.edu.ua/base/document?id=462>
7. Положення про академічну доброчесність учасників освітнього процесу Тернопільського національного технічного університету імені Івана Пулюя
URL: <https://docs.tntu.edu.ua/base/document?id=465>
8. Стандарт вищої освіти України: другий (магістерський) рівень, галузь знань 12 – Інформаційні технології, спеціальність – 121 Інженерія програмного забезпечення. Затверджено і введено в дію наказом Міністерства освіти і науки України № 1424 від 17.11.2020.
URL: <https://osvita.ua/doc/files/news/775/77548/121-inzheneriya-prohramnoho-zabezpechenn.pdf>
9. Освітньо-професійна програма “Інженерія програмного забезпечення” другого рівня вищої освіти за спеціальністю F2 “Інженерія програмного забезпечення”, галузі знань F2 “Інформаційні технології”. Режим доступу -
URL: <https://kaf-pi.tntu.edu.ua/programs-b/>
10. Положення про оцінювання здобувачів вищої освіти Тернопільського національного технічного університету імені Івана Пулюя (нова редакція) - наказ №4/7-670 від 25.09.2020

- URL: <https://docs.tntu.edu.ua/base/document?id=86>
11. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0 / ed. H. Washizaki. IEEE Computer Society, 2024. 411 p. URL: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf>
 12. Приклад структури репозиторію для коду [Електронний ресурс] URL: <https://github.com/TNTU-121-Software-Engineering/2024-2025-KRB-SE-41-Ivan-PULUJ>
 13. ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».
 14. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».
 15. Лист Міністерства освіти та науки України Зкладам вищої освіти та науковим установам «Про використання джерел інформації, створених на території держави-агресора» №1/19894-24 від 24.10.2024. URL: <https://rada.kpi.ua/files/lyst-mon-1-19894-24-vid-25-10-2024.pdf>
 16. ELARTU: Інституційний репозитарій ТНТУ. Роботи студентів. URL: <https://elartu.tntu.edu.ua/handle/123456789/31>
 17. Матеріали дистанційного курсу Електронного навчання ТНТУ з дисципліни «Раціональний уніфікований процес проєктування програмного забезпечення». ID курсу: 1237 URL: <https://dl.tntu.edu.ua/index.php>
 18. IEEE Editorial Style Manual for Authors URL: <https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Editorial-Style-Manual-for-Authors.pdf>
 19. Мартін Р. Чиста архітектура. Мистецтво розробки програмного забезпечення./ Пер. з англ. І. Боднр-Терещенко. Харків, 2022. 368 с.
 20. Фрімен Е., Робсон Е., Бейтс Б., Сієрра К. Head First. Патерни проєктування / пер. з англ. Г. Якубовської. 2-ге вид. Харків : Фабула, 2022. 672 с. (Серія «#PROSystem»).
 21. Гамма Е., Гелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проєктування. Патерни проєктування. Київ : Основи, 2019. 368 с.
 22. Швець О. Refactoring.Guru. Патерни проєктування та рефакторинг. URL: <https://refactoring.guru/uk/design-patterns> (дата звернення: 03.08.2025).
 23. Макконнелл С. Досконалий код / пер. з англ. Київ : Фабула, 2019. 896 с.
 24. ДСТУ ISO/IEC/IEEE 42010:2016 (ISO/IEC/IEEE 42010:2011, IDT). Системна та програмна інженерія. Опис архітектури. Київ : ДП «УкрНДНЦ», 2019.

- 25.ДСТУ ISO/IEC 25010:2016 (ISO/IEC 25010:2011, IDT). Інженерія систем і програмного забезпечення. Вимоги до якості систем і програмного забезпечення та оцінювання (SQuaRE). Моделі якості систем і програмного забезпечення. Київ : ДП «УкрНДНЦ», 2019.
- 26.ДСТУ ISO/IEC 12207:2017 (ISO/IEC 12207:2008, IDT). Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Київ : ДП «УкрНДНЦ», 2019.
- 27.ДСТУ ISO/IEC 19505-1:2015 (ISO/IEC 19505-1:2012, IDT). Інформаційні технології. Об'єктна група керування. Уніфікована мова моделювання (OMG UML). Частина 1. Інфраструктура. Київ : ДП «УкрНДНЦ», 2016.
- 28.ДСТУ ISO/IEC 19505-2:2015 (ISO/IEC 19505-2:2012, IDT). Інформаційні технології. Об'єктна група керування. Уніфікована мова моделювання (OMG UML). Частина 2. Суперструктура. Київ : ДП «УкрНДНЦ», 2016.

ДОДАТКИ

ДОДАТОК А

Зразок бланку титульної сторінки

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

КУРСОВИЙ ПРОЄКТ

з дисципліни

**«Рациональний уніфікований процес проєктування програмного
забезпечення»**

на тему:

**«Розробка архітектури застосунку для оцінки на антиплагіат наукових
робіт в використанні сучасних інформаційних технологій»**

Виконав: студент/ка III курсу, групи СП-__
спеціальності 121 Інженерія програмного забезпечення

_____ (підпис)

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

_____ (прізвище та ініціали)

Завідувач кафедри

_____ (підпис)

_____ (прізвище та ініціали)

Члени комісії

_____ (підпис)

_____ (прізвище та ініціали)

Оцінка за національною шкалою _____

Кількість балів: _____ Оцінка ECTS _____

Тернопіль 2025

ДОДАТОК Б

Зразок бланку завдання (двосторонній друк)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Кафедра _____

Дисципліна _____

Спеціальність _____

Курс _____ Група _____ Семестр _____

ЗАВДАННЯ на курсовий проєкт

Студентові _____
(прізвище, ім'я, по батькові)

1. Тема проєкту _____

2. Термін здачі студентом закінченого проєкту _____

3. Вихідні дані до проєкту _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) _____

5. Перелік графічного (ілюстративного) матеріалу, якщо передбачено _____

6. Дата видачі завдання _____

ДОДАТОК В

Типові приклади тем для виконання курсових проєктів здобувачами спеціальності F2 «Інженерія програмного забезпечення» з предмету «Рациональний уніфікований процес проєктування програмного забезпечення»:

Напрямок 1: Розподілені та хмарні системи (Cloud-native & Microservices)

У цих темах магістр має застосувати RUP для декомпозиції складної предметної області на незалежні компоненти/сервіси.

1. **Об'єктно-орієнтоване проєктування мікросервісної архітектури системи управління логістичними ланцюгами (Supply Chain Management).**
2. **Архітектурне моделювання мультитенантної (Multi-tenant) SaaS-платформи для електронного документообігу підприємства.**
3. **Проєктування розподіленої системи агрегації та обробки телеметричних даних у реальному часі.**
4. **Розробка архітектури високонавантаженого API-шлюзу (API Gateway) для інтеграції корпоративних сервісів.**
5. **Архітектурне проєктування хмарної системи управління відеоконтентом із застосуванням патерну CQRS (Command Query Responsibility Segregation).**
6. **Моделювання архітектури Event-Driven системи для процесингу фінансових транзакцій.**

Напрямок 2: Інтелектуальні системи та обробка великих даних (AI & Big Data)

Тут акцент робиться на проєктуванні пайплайнів обробки даних, абстрагуванні ML-моделей та інтеграції AI в класичне ПЗ.

7. **Проєктування архітектури системи підтримки прийняття медичних рішень на базі алгоритмів машинного навчання.**
8. **Архітектурне моделювання рекомендаційної системи електронної комерції з використанням графових баз даних.**
9. **Об'єктно-орієнтоване проєктування програмного ядра системи семантичного аналізу та обробки природної мови (NLP-модуль).**
10. **Розробка архітектури розподіленої системи збору та попередньої обробки Big Data для предиктивної аналітики обладнання.**
11. **Проєктування мікросервісу розпізнавання патернів аномальної поведінки користувачів (Fraud Detection System).**

Напрямок 3: Реінжиніринг, рефакторинг та еволюція ПЗ

Ідеально для магістрів, які вже працюють. Завдання — взяти моноліт або застарілу систему і проєктувати її перехід на нову архітектуру за допомогою RUP.

12. **Реінжиніринг та архітектурне проєктування переходу монолітної банківської системи на мікросервісну архітектуру.**
13. **Зворотне проєктування (Reverse Engineering) та архітектурна оптимізація існуючої CRM-системи підприємства.**

14. **Проектування прошарку інтеграції (Facade/Adapter) для модернізації застарілих (Legacy) корпоративних інформаційних систем.**
15. **Архітектурний рефакторинг високонавантаженого веб-додатка з метою підвищення його масштабованості та відмовостійкості.**

Напрямок 4: Відмовостійкі та безпекоорієнтовані системи (Resilient & Secure Systems)

Фокус на нефункціональних вимогах (FURPS+), які є критичними на фазі Elaboration у RUP.

16. **Проектування архітектури відмовостійкого платіжного шлюзу з підтримкою розподілених транзакцій (патерн Saga).**
17. **Архітектурне моделювання системи управління ідентифікацією та доступом (IAM) на основі протоколів OAuth 2.0 / OpenID Connect.**
18. **Розробка архітектури децентралізованої системи зберігання та верифікації цифрових контрактів (із застосуванням технологій DLT).**
19. **Проектування системи моніторингу та запобігання кібератакам на корпоративну інфраструктуру (Intrusion Detection System).**

Напрямок 5: Системи реального часу, IoT та спеціалізоване ПЗ

Теми, що вимагають моделювання складних станів (State Machine) та взаємодії апаратного і програмного забезпечення.

20. **Об'єктно-орієнтоване проектування ядра керування гетерогенною мережею IoT-пристроїв «Розумного міста».**
21. **Архітектурне моделювання системи диспетчеризації автономних роботизованих платформ (наприклад, для складських приміщень).**
22. **Проектування архітектури програмного комплексу для стрімінгу відеоданих із динамічною адаптацією бітрейту.**
23. **Розробка архітектурної моделі системи високочастотного трейдингу (HFT) з мінімізацією затримок обробки подій.**
24. **Проектування кросплатформного ядра геоінформаційної системи (ГІС) для моніторингу рухомих об'єктів.**

Напрямок 6: Інструментальне ПЗ та автоматизація (Software Engineering Tools)

Створення інструментів для самих розробників. Глибоко відповідає суті спеціальності 121.

25. **Архітектурне проектування системи оркестрації та автоматизації CI/CD пайплайнів.**
26. **Розробка архітектури спеціалізованого CASE-засобу для автоматичної генерації коду на основі UML-діаграм класів.**
27. **Проектування системи автоматизованого тестування (QA Automation Framework) для мікросервісних архітектур.**
28. **Архітектурне моделювання платформи агрегації та аналізу логів розподілених застосунків.**

ДОДАТОК Г

Розгорнутий зміст КП
спеціальності F2 «Інженерія програмного забезпечення»

АКТУАЛЬНИЙ ПРИКЛАД ЗМІСТУ КУРСОВОГО ПРОЄКТУ (РІВЕНЬ МАГІСТРА)

ВСТУП

РОЗДІЛ 1. БІЗНЕС-МОДЕЛЮВАННЯ ТА ІНЖЕНЕРІЯ ВИМОГ (ФАЗА INSERTION)

- 1.1. Аналіз предметної області та актуальність проєкту
- 1.2. Формування бачення проєкту (Vision) та визначення меж програмної системи
- 1.3. Бізнес-моделювання та виявлення зацікавлених осіб (Stakeholders)
- 1.4. Моделювання функціональних вимог системи
 - 1.4.1. Побудова UML-діаграми варіантів використання (Use Case Diagram)
 - 1.4.2. Специфікація та структурований опис критичних варіантів використання
- 1.5. Аналіз нефункціональних вимог та обмежень за моделлю FURPS+ (Supplementary Specification)

РОЗДІЛ 2. АРХІТЕКТУРНЕ ПРОЄКТУВАННЯ СИСТЕМИ (ФАЗА ELABORATION)

- 2.1. Вибір та обґрунтування архітектурного стилю і технологічного стеку
- 2.2. Логічний вигляд (Logical View)
 - 2.2.1. Декомпозиція системи на підсистеми та пакети (Package Diagram)
 - 2.2.2. Побудова концептуальної UML-діаграми класів предметної області
- 2.3. Процесний вигляд (Process View)
 - 2.3.1. Моделювання паралелізму, синхронізації та станів системи (State Machine / Activity Diagrams)
- 2.4. Вигляд реалізації (Development View)
 - 2.4.1. Проєктування компонентної бази та інтерфейсів взаємодії (Component Diagram)
 - 2.4.2. Обґрунтування застосованих патернів проєктування (GoF, GRASP) для забезпечення стійкості архітектури
- 2.5. Вигляд розгортання (Physical/Deployment View)
 - 2.5.1. Топологія апаратного забезпечення та розподіл компонентів (Deployment Diagram)

РОЗДІЛ 3. ДЕТАЛЬНЕ ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ (ФАЗИ CONSTRUCTION TA TRANSITION)

3.1. Детальне проектування взаємодії об'єктів для реалізації ключових сценаріїв (Sequence / Communication Diagrams)

3.2. Проектування моделі даних (Data Model) та механізмів збереження

3.3. Розробка базового архітектурного прототипу (Executable Architecture) та генерація коду в CASE-засобах

3.4. Стратегія забезпечення якості та планування тестування (Test Plan)

3.4.1. Розробка тестових сценаріїв для перевірки архітектурних рішень

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ (Специфікації вимог, лістинги критичних ділянок коду, розширені UML-моделі)