

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

Кафедра програмної інженерії



МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсового проєкту з дисципліни
**«Об'єктні технології та інструментальні засоби
конструювання 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

ЛІТЕРАТУРА

НАВЧАЛЬНО-МЕТОДИЧНА

Методичні вказівки до виконання курсового проєкту з дисципліни «Об'єктні технології та інструментальні засоби конструювання 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:

Михалик Д.М. к.т.н., доцент каф. ПІ
Цебрій О.Р., к.ф.-м.н., доцент каф. ПІ
Стефанишин В.М., ас. каф. ПІ

Рецензент:

Загородна Н. В., к.т.н, доцент,
завідувач кафедрою кібербезпеки ТНТУ ім. Івана Пулюя

Відповідальний за випуск: ас. каф. ПІ, Стефанишин В.М.

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

Протокол № 1 від 29 серпня 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 1 вересня 2025 р.

Методичні вказівки до виконання курсового проєкту з дисципліни «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення» для здобувачів другого «магістерського» рівня освіти за спеціальністю F2 – «Інженерія програмного забезпечення», всіх форм навчання / укладачі: Михалик Д.М., Цебрій О.Р., Стефанишин В.М. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025. – 49 с.

Методичні вказівки призначені для ознайомлення з вимогами, правилами оформлення та оцінювання курсових проєктів для здобувачів вищої освіти другого «магістерського» рівня освіти, 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 «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення». В методичних вказівках розкрито суть та загальні положення щодо виконання курсового проєктування; 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	6
2 МЕТА Й ЗАВДАННЯ	9
3. ОРГАНІЗАЦІЯ ВИКОНАННЯ	11
3.1 Підготовчий етап	13
3.2 Основний етап	14
3.3 Заключний етап	15
3.4 Права та обов'язки здобувача	16
4 ВИМОГИ ДО ЗМІСТУ ТА СТРУКТУРИ	18
5 ВИМОГИ ДО ОФОРМЛЕННЯ	21
5.1 Загальні вимоги	21
5.2 Вимоги до обов'язкових структурних елементів	24
5.3 Оформлення списків	25
5.4 Оформлення ілюстрацій	26
5.5 Оформлення таблиць	27
5.6 Оформлення формул та рівнянь	28
5.7 Оформлення лістингів програмного коду	29
5.8 Оформлення посилань на літературні джерела	30
5.9 Оформлення додатків	31
5.10 Оформлення коду та супровідної документації	32
6 ПОРЯДОК ПЕРЕВІРКИ ТА ЗАХИСТУ	33
6.1 Підготовка до захисту	33
6.2 Академічна доброчесність	34
6.3 Підготовка презентації та доповіді	35
6.4 Порядок захисту курсового проєкту	37
6.5 Оцінювання курсових проєктів	38
СПИСОК ДЖЕРЕЛ	40
ДОДАТКИ	42
ДОДАТОК А	43
ДОДАТОК Б	44
ДОДАТОК В	46
ДОДАТОК Г	48

ВСТУП

Курсовий проєкт з дисципліни «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення» є ключовим етапом підготовки магістрів спеціальності F2 «Інженерія програмного забезпечення». Він спрямований на інтеграцію теоретичних знань та формування поглиблених практичних компетенцій у напрямку проєктування та конструювання складних програмних систем.

Основні завдання курсового проєкту полягають у:

- Аналізі існуючого програмного проєкту: дослідженні архітектури, структури програмного коду, використаних технологій та інструментів, а також оцінці відповідності реалізації сучасним принципам об'єктно-орієнтованого програмування, стандартам та кращим практикам розробки.
- Виявленні проблем та обмежень: ідентифікації недоліків у проєктуванні та реалізації програмного забезпечення, зокрема порушень принципів SOLID, проблем модульності, масштабованості, повторного використання коду та підтримованості системи.
- Проєктуванні покращень: розробці обґрунтованих рішень щодо вдосконалення програмного продукту із застосуванням сучасних підходів, шаблонів проєктування, архітектурних принципів та технологій.
- Моделюванні програмної системи: побудові UML-діаграм (класів, компонентів, взаємодії тощо) для відображення існуючої та вдосконаленої архітектури системи, опису структури та поведінки програмних компонентів.
- Рефакторингу та оптимізації: виконанні змін у програмному коді з метою покращення його структури, читабельності, ефективності та відповідності сучасним вимогам до якості програмного забезпечення.
- Використанні інструментальних засобів: застосуванні сучасних середовищ розробки, систем контролю версій, засобів аналізу коду та інших інструментів, що підтримують процес конструювання програмного забезпечення.
- Забезпеченні якості програмного забезпечення та обґрунтуванні результатів: перевірці та оцінці змін із використанням методів тестування, аналізу якості коду та інших підходів для підтвердження ефективності внесених покращень; проведенні аналізу впливу внесених змін на продуктивність, гнучкість, масштабованість і підтримованість програмного продукту, а також формуванні висновків щодо доцільності застосованих рішень.

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 конструювання 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 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 професійно-орієнтовану та науково-практичну підготовку, спрямовану на розв'язання складних задач аналізу, конструювання та вдосконалення програмних систем із використанням сучасних технологій, інструментальних засобів і підходів програмної інженерії.

Курсовий проєкт з дисципліни «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення» для здобувачів другого (магістерського) рівня вищої освіти (надалі – КП) є самостійною практичною роботою. Він демонструє здатність здобувача виконувати глибокий аналіз існуючих програмних проєктів, оцінювати їх архітектурні рішення, якість реалізації та відповідність сучасним стандартам, а також розробляти і впроваджувати обґрунтовані покращення із застосуванням об'єктно-орієнтованих підходів, шаблонів проєктування та сучасних інструментальних засобів розробки.

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

Основним критерієм при виконанні проєкту є обґрунтованість прийнятих рішень, відповідність сучасним принципам конструювання програмного забезпечення та практична цінність запропонованих покращень. Особлива увага приділяється якості аналізу існуючого програмного продукту, глибині виявлених проблем та ефективності запропонованих змін.

Робота над проєктом базується на сучасних підходах програмної інженерії, що передбачає:

- Аналіз програмного продукту: дослідження існуючого програмного проєкту, його архітектури, структури коду та використаних технологій. Визначення функціональних і нефункціональних характеристик системи, виявлення проблемних місць, недоліків у проєктуванні та реалізації, а також оцінка відповідності сучасним принципам програмної інженерії.
- Проєктування (архітектурне вдосконалення): розробка обґрунтованих рішень щодо покращення системи із застосуванням сучасних архітектурних підходів та шаблонів проєктування. Формування нової або вдосконаленої структури системи, визначення взаємодії компонентів, підвищення модульності, масштабованості та підтримованості програмного продукту.
- Реалізація покращень: внесення змін у програмний код відповідно до розроблених рішень, виконання рефакторингу, оптимізація структури класів і компонентів, застосування об'єктно-орієнтованих принципів та сучасних інструментальних засобів розробки.

- Перевірка якості та аналіз ефективності: оцінка результатів впроваджених змін шляхом аналізу якості коду, тестування, вимірювання продуктивності та інших характеристик системи. Визначення впливу внесених покращень на ефективність, гнучкість, масштабованість і підтримуваність програмного забезпечення.

Дисциплінарна інтеграція: У курсовому проєкті з дисципліни «Об’єктні технології та інструментальні засоби конструювання програмного забезпечення» здобувачі вирішують комплексні інженерні задачі, що базуються на компетенціях, здобутих під час вивчення таких дисциплін, як: «Основи програмування»; «Об’єктно-орієнтоване програмування»; «Алгоритми і структури даних»; «Основи програмної інженерії»; «Аналіз вимог до програмного забезпечення»; «Моделювання та аналіз програмного забезпечення»; «Архітектура та проєктування програмного забезпечення»; «Бази даних»; «Якість програмного забезпечення та тестування»; «Конструювання програмного забезпечення»; «Управління ІТ-проєктами»; «Раціональний уніфікований процес проєктування програмного забезпечення».

Курсовий проєкт інтегрує набуті знання для комплексного аналізу та вдосконалення існуючого програмного продукту - від оцінки архітектурних рішень і структури коду до реалізації змін та перевірки їх ефективності відповідно до сучасних стандартів і практик розробки програмного забезпечення.

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 «Об’єктні технології та інструментальні засоби конструювання програмного забезпечення».

У цих методичних вказівках наведено систематизовані вимоги до організації процесу виконання курсового проекту, його змісту, логічної структури та технічного оформлення результатів роботи, орієнтованих на аналіз, проектування, реалізацію та вдосконалення програмного забезпечення відповідно до сучасних стандартів і практик розробки.

2 МЕТА Й ЗАВДАННЯ

Метою виконання курсового проєкту є демонстрація здобувачем здатності до комплексного та творчого розв'язання складних завдань аналізу, вдосконалення та конструювання 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 освітньо-професійній програмі другого (магістерського) рівня.

Основними завданнями курсового проєкту є:

- Систематизація та поглиблення компетенцій: закріплення та розширення теоретичних знань і практичних навичок об'єктно-орієнтованого програмування, архітектурного аналізу та конструювання програмного забезпечення, а також їх застосування при вирішенні задач вдосконалення існуючих програмних систем.
- Аналіз програмного продукту: набуття навичок дослідження архітектури, структури коду та технологічних рішень існуючих програмних проєктів, виявлення недоліків та оцінювання відповідності сучасним стандартам і принципам програмної інженерії.
- Архітектурне проєктування та реорганізація: формування вмінь обґрунтовувати та впроваджувати покращення архітектури програмного забезпечення із застосуванням сучасних підходів, принципів та шаблонів проєктування.
- Рефакторинг та реалізація покращень: розвиток навичок внесення змін у програмний код, оптимізації структури програмних компонентів, підвищення якості, гнучкості та підтримуваності програмного продукту.
- Інструментальна майстерність: ефективне використання сучасних середовищ розробки, систем контролю версій, засобів аналізу коду та інших інструментальних засобів для реалізації процесу вдосконалення програмного забезпечення.
- Професійна аналітика та оцінювання ефективності: розвиток здатності оцінювати результати внесених змін, аналізувати їх вплив на продуктивність, масштабованість та якість програмного забезпечення, а також обґрунтовувати прийняті інженерні рішення.

Виконання КП передбачає формування та розвиток у студентів компетентностей:

інтегральна компетентність:

- Здатність 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 і вимог.

Загальні компетентності:

- ЗК01 Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК03 Здатність проводити дослідження на відповідному рівні.
- ЗК05. Здатність генерувати нові ідеї (креативність)

Спеціальні (фахові) компетентності:

- СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.
- СК03. Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.
- СК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування учасних моделей, методів та технологій розроблення програмного забезпечення.
- СК09. Здатність забезпечувати якість програмного забезпечення.
- СК10*. Здатність розробляти інноваційне програмне забезпечення.

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

- РН02 Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.
- РН06 Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.
- РН08 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
- РН09 Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.
- 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.

Магістри мають змогу виконувати завдання, пов'язані з аналізом, проєктуванням, оптимізацією та супроводом складних програмних систем у різних сферах, обіймати посади архітекторів програмних систем (Software Architect), керівників команд розробки (Team Lead), інженерів-програмістів (Software Engineer), інженерів із забезпечення якості (QA Engineer), DevOps-інженерів, а також фахівців з аналізу та обробки даних відповідно до Національного класифікатора професій ДК 003:2010.

Курсовий проєкт є самостійно виконаною і належним чином оформленою роботою здобувача з дисципліни «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення», що передбачає вирішення практичного завдання аналізу існуючого програмного продукту та впровадження обґрунтованих покращень на основі набутих знань і практичних навичок.

3. ОРГАНІЗАЦІЯ ВИКОНАННЯ

Формування тематики та змісту завдань

Завдання на курсовий проєкт має бути орієнтоване на розв'язання прикладних інженерних задач аналізу, проєктування та вдосконалення існуючого програмного забезпечення. Основою роботи є дослідження реального програмного проєкту з відкритим кодом, оцінка його архітектури, структури та якості реалізації, а також впровадження обґрунтованих покращень із використанням сучасних принципів об'єктно-орієнтованого програмування, архітектурних підходів, шаблонів проєктування та інструментальних засобів розробки. Магістерський рівень передбачає не лише застосування відомих підходів, а й їх адаптацію до конкретних умов існуючого проєкту з урахуванням його обмежень, технологічного стеку та особливостей реалізації.

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).

Пропоновані тематики 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.

Обрана тема курсового проєкту повинна бути актуальною, мати обґрунтовану практичну цінність та відповідати сучасному стану розвитку інженерії програмного забезпечення. Вона має передбачати аналіз існуючого програмного продукту з відкритим кодом і можливість його змістовного вдосконалення із застосуванням сучасних підходів, архітектурних рішень, шаблонів проєктування та інструментальних засобів розробки.

Проєкт повинен відображати сучасні тенденції розробки програмного забезпечення, включати використання актуальних технологій, принципів об'єктно-орієнтованого програмування, підходів до забезпечення якості та супроводу програмних систем. Особлива увага приділяється обґрунтованості запропонованих змін та їх відповідності кращим практикам і стандартам галузі.

Тема курсового проєкту повинна забезпечувати можливість проведення повного циклу робіт: аналізу програмного продукту, проєктування покращень, їх реалізації та оцінювання ефективності отриманих результатів.

Тематика курсових проєктів, рекомендована в межах спеціальності (див. Додаток Г), орієнтована на розв'язання складних інженерних задач аналізу, проєктування та вдосконалення існуючого програмного забезпечення із застосуванням сучасних методів моделювання та інструментальних засобів розробки і спрямована на:

1. Вдосконалення архітектурних рішень: аналіз існуючих систем та покращення їх архітектури з урахуванням сучасних підходів (модульність, масштабованість, сервісна або мікросервісна організація), застосування актуальних архітектурних патернів.
2. Реінжиніринг та модернізацію ПЗ: дослідження наявного програмного коду, виконання рефакторингу, зворотного проєктування (Reverse Engineering), а також адаптація систем до сучасних технологічних стеків або вимог.
3. Покращення якості програмного забезпечення: впровадження практик забезпечення якості, тестування, аналізу коду, оптимізації продуктивності та підвищення надійності програмних систем.
4. Використання інструментальних засобів: застосування сучасних середовищ розробки, систем контролю версій, засобів автоматизації (CI/CD), аналізу коду та інших інструментів для ефективного вдосконалення програмного продукту.
5. Розширення функціональності систем: розробка нових модулів, компонентів або сервісів для існуючих програмних продуктів, інтеграція з іншими системами через API та інші механізми взаємодії.
6. Оптимізацію та підвищення ефективності: покращення продуктивності, масштабованості та використання ресурсів програмного забезпечення на основі аналізу існуючих реалізацій.
7. Адаптацію до сучасних вимог: приведення програмного продукту у відповідність до сучасних стандартів, практик та тенденцій розробки програмного забезпечення, включаючи використання сучасних мов програмування, бібліотек і фреймворків.

Організаційно процес виконання курсового проєкту інтегровано з підходом поетапного вдосконалення програмного забезпечення і складається з трьох ключових макроетапів:

- **Підготовчий етап (аналіз):** вибір програмного проєкту, дослідження предметної області, аналіз архітектури, структури коду та використаних технологій. Визначення проблемних місць, формування вимог до вдосконалення системи та вибір інструментальних засобів для реалізації змін.
- **Основний етап (проєктування та реалізація):** розробка обґрунтованих рішень щодо покращення архітектури та структури програмного

забезпечення, побудова UML-моделей (за потреби), виконання рефакторингу та реалізація змін у програмному кодї з використанням сучасних підходів і технологій.

- **Заключний етап (оцінка та оформлення):** перевірка якості реалізованих змін, тестування та аналіз ефективності вдосконалень, формування пояснювальної записки, підготовка презентаційних матеріалів і публічний захист курсового проєкту.

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

3.1 Підготовчий етап

Під час підготовчого етапу здобувач обирає програмний проєкт з відкритим кодом, узгоджує його з керівником та отримує індивідуальне завдання щодо аналізу та напрямів вдосконалення. На цьому етапі здійснюється ознайомлення з функціональністю системи, її архітектурою, структурою програмного коду та використаними технологіями. Проводиться аналіз предметної області, визначаються основні проблеми та обмеження програмного продукту, формуються вимоги до його покращення, а також виконується огляд сучасних підходів, технологій і практик, що можуть бути застосовані для вдосконалення системи.

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

Вибір напрямку та керівника: на початку семестру (відповідно до графіка навчального процесу) здобувач обирає програмний проєкт та керівника зі складу викладачів випускової кафедри.

Керівником курсового проєкту може бути науково-педагогічний працівник кафедри або кваліфікований фахівець-практик ІТ-галузі за профілем спеціальності. Обраний проєкт повинен бути достатньо складним і містити можливості для проведення повноцінного аналізу та впровадження суттєвих покращень.

У разі, якщо проєкт є масштабним або багатокomпонентним, допускається виконання комплексного курсового проєкту кількома здобувачами. При цьому між учасниками розподіляються ролі та відповідальність, зокрема аналіз архітектури, проєктування покращень, реалізація змін та оцінювання якості програмного продукту.

Формування індивідуального завдання (Vision): Спільно з керівником формується індивідуальне завдання, яке визначає межі дослідження та вдосконалення обраного програмного проєкту. Завдання охоплює: аналіз функціональності, архітектури та структури, збирання фактичних матеріалів, проведення системного аналізу, а також визначення напрямів і вимог до покращення (функціональних та нефункціональних).

Завдання на курсовий проєкт:

Тема курсового проєкту затверджується керівником на початку семестру. Відповідно до обраного програмного проєкту студент разом із керівником формує індивідуальне завдання. У завданні зазначається: тема курсового проєкту; термін подання завершеної роботи; початкові дані; зміст роботи (перелік питань, що підлягають розробці). Завдання підписується керівником і студентом.

У завданні передбачається:

- п.1 – формулювання теми курсового проєкту, яка повинна чітко відображати суть аналізу та вдосконалення обраного програмного продукту;
- п.2 – визначення терміну подання завершеної роботи на перевірку;
- п.3 – формулювання мети роботи, короткий опис програмного проєкту, що аналізується, постановка задач щодо його вдосконалення, визначення типу змін (рефакторинг, розширення функціональності, оптимізація, інтеграція тощо);
- п.4 – визначення структури пояснювальної записки (основні розділи: аналіз системи, проєктування покращень, реалізація, оцінка ефективності);
- п.5 – зазначення технічних вимог: використані технології, мова програмування, середовище розробки, системи контролю версій, інструменти аналізу та тестування, формат вхідних і вихідних даних;
- п.6 – дата затвердження завдання на курсовий проєкт.

Форма завдання на курсовий проєкт наведена у відповідному додатку.

3.2 Основний етап

Під час основного етапу магістрант виконує детальний аналіз обраного програмного проєкту, розробляє обґрунтовані рішення щодо його вдосконалення, реалізує необхідні зміни у програмному коді та готує результати для перевірки керівником. Основна увага приділяється якості архітектурних рішень, коректності реалізації та ефективності внесених покращень.

Обов'язки керівника КП:

- Фахове консультування: надає здобувачеві рекомендації щодо аналізу архітектури програмного продукту, застосування принципів об'єктно-орієнтованого програмування, шаблонів проєктування та сучасних підходів до вдосконалення програмного забезпечення.
- 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 виконання етапів роботи, своєчасність реалізації поставлених завдань та коректність прийнятих рішень.

- Контроль дисципліни: у разі суттєвих порушень графіку, 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 проєктування.
- Архітектурні рев'ю: проводить регулярні консультації, під час яких здобувач звітує про виконання роботи, обговорюються варіанти вдосконалення, перевіряється узгодженість архітектурних рішень та якість програмного коду..
- Підготовка до захисту: консультує щодо структури презентації та правильного оформлення Документа архітектури програмного забезпечення (Software Architecture Document).

На цьому етапі всі результати роботи (аналіз системи, проєктні рішення, реалізовані зміни та оцінка ефективності) повинні бути повністю завершені, оформлені у вигляді пояснювальної записки та узгоджені з керівником.

3.3 Заключний етап

Під час заключного етапу відбувається фіналізація всіх результатів курсового проєктування, їх оформлення та підготовка до публічного захисту. Основна увага приділяється узгодженості виконаної роботи, якості реалізованих змін і обґрунтованості отриманих результатів.

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 затвердження 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 студентських 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). Проект вважається повністю готовим до захисту після отримання всіх необхідних підписів (здобувача, керівника) на титульному аркуші та в завданні на курсовий проект. Також передбачається здача електронної версії артефактів (файли моделей з CASE-засобів, текст пояснювальної записки, презентація) до архіву кафедри.

3.4 Права та обов'язки здобувача

Здобувач **зобов'язаний** [3–8]:

- Організація роботи: своєчасно обрати керівника з числа науково-педагогічних працівників кафедри програмної інженерії, узгодити з ним тему курсового проекту, яка передбачає аналіз та вдосконалення існуючого програмного продукту, та отримати індивідуальне завдання.
- Планування (ітераційний підхід): скласти та узгодити з керівником календарний план виконання курсового проекту, що передбачає поетапне виконання робіт: аналіз, проектування, реалізація та оцінка ефективності. Регулярно звітувати про результати кожного етапу.
- Самостійність: самостійно виконувати всі етапи курсового проекту або визначену частину роботи у разі командного виконання, забезпечуючи повне розуміння обраного програмного продукту та внесених змін.
- Науково-технічний рівень: застосовувати сучасні підходи інженерії програмного забезпечення, зокрема принципи об'єктно-орієнтованого програмування (ООП), об'єктно-орієнтованого аналізу та дизайну (OOAD), принципи SOLID, шаблони проектування (GoF), а також сучасні архітектурні підходи (layered architecture, microservices, clean architecture).
- Аналіз програмного продукту: провести детальний аналіз існуючої системи, включаючи її архітектуру, структуру коду, взаємодію компонентів, використані технології, а також визначити проблемні місця та обґрунтувати необхідність їх покращення.
- Проектування вдосконалень: розробити обґрунтовані рішення щодо покращення програмного продукту, включаючи зміни архітектури, структури класів, взаємодії компонентів, із використанням UML-діаграм (class, sequence, component, deployment) відповідно до специфікації UML 2.5.
- Реалізація: реалізувати запропоновані зміни з використанням сучасних мов програмування та технологій, забезпечуючи відповідність коду принципам чистого коду (Clean Code), рефакторингу та повторного використання компонентів.
- Оцінка якості та ефективності: провести аналіз результатів впроваджених змін, включаючи оцінку якості програмного забезпечення (maintainability, scalability, performance), використовуючи метрики якості коду (наприклад, cyclomatic complexity, code smells) та інструменти статичного аналізу.
- Використання інструментальних засобів: застосовувати сучасні засоби розробки та супроводу ПЗ, зокрема сервіси контролю версій (Bitbucket, GitHub, GitLab), середовища розробки (IntelliJ IDEA, Visual Studio, VS Code), засоби моделювання (draw.io, StarUML, Visual Paradigm), а також інструменти аналізу якості коду (SonarQube, ESLint, Pylint).

- Відповідальність за якість: нести персональну відповідальність за коректність виконаного аналізу, обґрунтованість запропонованих змін, якість реалізації та відсутність академічного плагіату.
- Дотримання регламенту: дотримуватися встановленого графіка виконання курсового проєкту та своєчасно реагувати на зауваження керівника.
- Подання роботи: у встановлені терміни подати завершений курсовий проєкт на перевірку, після чого усунути зауваження та отримати допуск до захисту.
- Академічна доброчесність: забезпечити оригінальність виконаної роботи, пройти перевірку на плагіат відповідно до вимог закладу вищої освіти.
- Захист: своєчасно з'явитися на публічний захист курсового проєкту та представити результати виконаної роботи з обґрунтуванням прийнятих рішень.

Здобувач **має право** [3, 4, 5]:

- Вибір та адаптація теми: Обирати тему курсового проєкту 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 проєктом 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, проєктування архітектури та тестування розроблених моделей ПЗ.
- Фаховий супровід: Отримувати систематичні консультації керівника курсового проєкту згідно із затвердженим розкладом для обговорення результатів кожної ітерації, аналізу архітектурних ризиків та верифікації проєктних 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 проєктування програмних систем.

4 ВИМОГИ ДО ЗМІСТУ ТА СТРУКТУРИ

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 (двосторонній 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 існуючого програмного продукту та вимоги до вдосконалення (аналіз архітектури, коду, технологій, визначення проблем та формування вимог)	5-10	+

Розділ 2 Проєктування вдосконалень та архітектурних рішень (рефакторинг архітектури, UML-моделі, застосування патернів, обґрунтування змін)	10-15	+
Розділ 3 Реалізація змін та оцінка якості програмного забезпечення (впровадження покращень, тестування, аналіз ефективності, метрики якості)	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 заголовку 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

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, аббревіатури, 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 проекту (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 (рисунки, 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 – XXXXXXXXXXXX» – перша таблиця другого розділу, «Таблиця В.2 – XXXXXXXXXXXX» – 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 на 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: $(65 \pm 3)\%$; $80 \text{ м} \pm 2 \text{ м}$ 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: $80 \text{ м} \times 25 \text{ м} \times 50 \text{ м}$ (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., Біщак 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 GIT репозиторію та розмістити його в GitHub організації випускової кафедри (<https://github.com/orgs/TNTU-121-Software-Engineering>). Приклад назви репозиторію 2025-2026-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 (КП), підписаний здобувачем вищої освіти, подається на первинну перевірку керівникові у вигляді електронної версії пояснювальної записки та незброшурованої паперової чернетки (за вимогою).

До пояснювальної записки обов'язково додається структурований електронний архів або посилання на репозиторій (GitHub, GitLab або інший), який містить матеріали, що підтверджують виконання всіх етапів курсового проєкту: аналізу, проєктування, реалізації та оцінки ефективності змін.

Електронні артефакти повинні включати:

- Файли моделей та проєктних рішень: вихідні файли UML-моделей та інших схем (наприклад, draw.io, StarUML, Visual Paradigm), що відображають як існуючу архітектуру системи, так і запропоновані вдосконалення (class, sequence, component, deployment діаграми).
- Документацію: електронну версію пояснювальної записки, а також додаткові матеріали, що описують результати аналізу системи, сформульовані вимоги до покращення, обґрунтування архітектурних рішень та опис внесених змін.
- Програмну реалізацію: вихідний код програмного продукту (або його частини), що включає як початкову версію (за можливості), так і реалізовані вдосконалення. Код повинен бути структурованим, зрозумілим та відповідати принципам чистого коду і сучасним практикам розробки.
- Контроль версій: історію змін у системі контролю версій (Git), що демонструє процес розвитку проєкту, внесення змін, рефакторинг та етапність виконання роботи.
- Інфраструктурні та конфігураційні файли (за наявності): скрипти запуску проєкту, файли конфігурації, налаштування середовища, а також, за необхідності, файли для контейнеризації (Docker) або розгортання системи.

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 рев'ю 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 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 без належних посилань), фабрикацію або фальсифікацію результатів;
- у яких відсутній аналіз існуючого програмного продукту або не обґрунтовано запропоновані покращення;
- що не містять реалізованих змін або їх результати не підтверджені аналізом якості та ефективності.

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 будь яких 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.)

6.3 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 існуючого програмного продукту;
- постановку задачі (мета та задачі курсового проєкту, що саме покращується);
- аналіз архітектури та проблемних місць системи;
- специфікацію вимог до вдосконалення (функціональні та нефункціональні);
- UML-діаграму варіантів використання (As-Is / To-Be) з короткими описами;
- діаграму структури класів або компонентів існуючої та оновленої системи;
- діаграми класів для ключових покращених частин системи;
- діаграми послідовностей, що відображають основні сценарії роботи до і після змін;
- діаграми діяльності, що демонструють бізнес-логіку або алгоритми після вдосконалення;
- (за потреби) діаграми станів для складних об'єктів;
- реалізовані зміни у програмному коді (фрагменти або структура);
- результати тестування та порівняння до/після (якість, продуктивність, інші метрики);
- використані інструментальні засоби (IDE, Git, засоби аналізу, UML-інструменти).

Завершальні слайди презентації повинні містити:

- отримані результати вдосконалення системи та їх практичне значення;
- обґрунтування ефективності запропонованих рішень;
- висновки за результатами виконання курсового проєкту.

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 рекомендовано принаймні один раз проговорити доповідь вголос, одночасно запустивши презентацію, оцінити (а при потребі скоригувати) її час, 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)	2026_KR_OT IKPZ_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_2026.pdf
Презентація роботи у форматі .pdf з текстовим шаром, вбудованими шрифтами та кодування UTF-8.	2026_KP_OT IKPZ-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..	2026_KP_OT IKPZ-SE-5-Іван-Павлович-ПУЛЮЙ-антиплагіат.pdf
Програма, що виконується (усі файли із модулями, що виконуються, а також усі необхідні бібліотеки якщо вони не публічно доступні). Каталог заархівований у форматі .zip без захисту паролем. Рекомендовано GIT репозиторій.	2026_KR_OT IKPZ_SE_51_I_PULUJ - project-src.zip
Усі додаткові файли, котрі не були включені раніше варто додати до архіву у форматі .zip без захисту паролем.	2026_KR_OT IKPZ_SE_51_I_PULUJ - project-docs.zip

6.4 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.:

6.5 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;
- проведено глибокий та обґрунтований аналіз існуючого програмного продукту;
- запропоновані архітектурні та структурні покращення є логічними, сучасними та повністю відповідають вимогам;
- реалізовані зміни покращують якість системи (підтверджено метриками або тестуванням);
- UML-моделі побудовані коректно, узгоджено та відображають як існуючий стан, так і внесені зміни;
- здобувач впевнено обґрунтовує прийняті рішення, володіє сучасними підходами (OOP, SOLID, патерни проєктування) та інструментами розробки.

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

- проєкт виконано належним чином, але є незначні недоліки в аналізі, проєктуванні або оформленні;
- аналіз системи проведено, проте не всі проблеми виявлені або обґрунтовані;
- архітектурні покращення є коректними, але частково обмежені або не повністю реалізовані;
- результати покращень підтверджені частково або поверхнево;
- під час захисту здобувач допускає незначні неточності у відповідях.

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

- робота містить суттєві недоліки в аналізі або відсутня глибина дослідження системи;
- запропоновані покращення є поверхневими або слабо обґрунтованими;
- архітектурні рішення спрощені або не відповідають сучасним підходам;
- реалізація змін є частковою або не підтверджена аналізом якості;
- здобувач відчуває труднощі під час пояснення прийнятих рішень.

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

- проєкт не відповідає поставленому завданню або тема не розкрита;
- відсутній аналіз існуючого програмного продукту або запропоновані зміни не мають обґрунтування;
- реалізація відсутня або непрацездатна;
- виявлено академічний плагіат або роботу виконано не самостійно;
- здобувач не володіє матеріалом і не може відповісти на базові запитання.

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.
9. URL: <https://osvita.ua/doc/files/news/775/77548/121-inzheneriya-prohramnoho-zabezpechenn.pdf>
10. 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/>
11. 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>
12. 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>
13. Приклад структури репозиторію для коду [Електронний ресурс] URL: <https://github.com/TNTU-121-Software-Engineering/2024-2025-KRB-SE-41-Ivan-PULUJ>
14. ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».

15. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».
16. 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>
17. ELARTU: Інституційний репозитарій ТНТУ. Роботи студентів. URL: <https://elartu.tntu.edu.ua/handle/123456789/31>
18. 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>
19. IEEE Editorial Style Manual for Authors URL: <https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Editorial-Style-Manual-for-Authors.pdf>
20. Мартін Р. Чиста архітектура. Мистецтво розробки програмного забезпечення./ Пер. з англ. І. Боднр-Терещенко. Харків, 2022. 368 с.
21. 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»).
22. Гамма Е., Гелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проєктування. Патерни проєктування. Київ : Основи, 2019. 368 с.
23. Швець О. Refactoring.Guru. Патерни проєктування та рефакторинг. URL: <https://refactoring.guru/uk/design-patterns> (дата звернення: 03.08.2025).
24. Макконнелл С. Досконалий код / пер. з англ. Київ : Фабула, 2019. 896 с.
25. ДСТУ ISO/IEC/IEEE 42010:2016 (ISO/IEC/IEEE 42010:2011, IDT). Системна та програмна інженерія. Опис архітектури. Київ : ДП «УкрНДНЦ», 2019.
26. 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.
27. ДСТУ ISO/IEC 12207:2017 (ISO/IEC 12207:2008, IDT). Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Київ : ДП «УкрНДНЦ», 2019.
28. 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.
29. 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

**«Об'єктні технології та інструментальні засоби конструювання
програмного забезпечення»**

на тему:

Виконав: студент/ка V курсу, групи СПм-51
спеціальності F2 Інженерія програмного забезпечення

Керівник Завідувач

(підпис)

(прізвище та ініціали)

кафедри

(підпис)

(прізвище та ініціали)

Члени комісії

(підпис)

(прізвище та ініціали)

Оцінка за національною шкалою

Кількість балів:

Оцінка ECTS

Тернопіль 2026

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 _____

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

(прізвище, ім'я, по батькові)

Керівник проєкту

(підпис)

(вчений ступінь, посада, прізвище, ім'я, по батькові)

ДОДАТОК В

Типові приклади тем для виконання курсових проєктів здобувачами спеціальності F2 «Інженерія програмного забезпечення» з предмету «Об'єктні технології та інструментальні засоби конструювання програмного забезпечення»:

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

У межах цього напрямку здобувач здійснює аналіз існуючих систем або їх прототипів і пропонує архітектурні покращення з використанням сучасних підходів.

1. **Аналіз та вдосконалення мікросервісної архітектури системи управління логістичними ланцюгами.**
2. **Рефакторинг архітектури SaaS-платформи електронного документообігу з урахуванням масштабованості та мультитенантності.**
3. **Аналіз та оптимізація системи обробки телеметричних даних у реальному часі.**
4. **Вдосконалення архітектури API-шлюзу для інтеграції корпоративних сервісів.**
5. **Рефакторинг хмарної системи управління відеоконтентом із використанням сучасних архітектурних підходів.**
6. **Аналіз та оптимізація Event-Driven системи обробки транзакцій.**

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

Основна увага приділяється інтеграції інтелектуальних компонентів у програмні системи та їх вдосконаленню.

7. **Аналіз та вдосконалення системи підтримки прийняття рішень із використанням машинного навчання.**
8. **Оптимізація архітектури рекомендаційної системи електронної комерції.**
9. **Рефакторинг програмного модуля обробки природної мови (NLP).**
10. **Аналіз та вдосконалення системи обробки даних для предиктивної аналітики.**
11. **Оптимізація модуля виявлення аномальної поведінки користувачів.**

Напрямок 3: Реінжиніринг, рефакторинг та еволюція ПЗ

Ключовий напрям для курсу — аналіз існуючого програмного продукту та його покращення.

12. **Рефакторинг архітектури монолітного веб-додатка з переходом до модульної або сервісної структури.**
13. **Зворотне проєктування (Reverse Engineering) та вдосконалення існуючої інформаційної системи.**
14. **Оптимізація структури коду та архітектури CRM-системи.**
15. **Реалізація патернів проєктування для модернізації застарілих систем (Facade, Adapter, Strategy).**
16. **Покращення масштабованості та підтримуваності існуючого програмного продукту.**

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

Фокус на нефункціональних характеристиках програмного забезпечення.

- 17. Аналіз та вдосконалення відмовостійкості програмної системи.
- 18. Оптимізація продуктивності високонавантаженого веб-додатка.
- 19. Рефакторинг системи автентифікації та авторизації користувачів.
- 20. Аналіз та покращення безпеки програмного забезпечення.
- 21. Вдосконалення системи моніторингу та логування.

Напрямок 5: Системи реального часу, IoT та спеціалізоване ПЗ

Передбачає роботу з системами, що мають складну логіку та обмеження.

- 22. Аналіз та оптимізація програмного забезпечення для IoT-систем.
- 23. Рефакторинг системи обробки подій у реальному часі.
- 24. Вдосконалення програмного комплексу обробки потокових даних.
- 25. Оптимізація системи керування ресурсами або процесами.
- 26. Аналіз та покращення алгоритмів обробки даних у спеціалізованому ПЗ.

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

Орієнтація на створення або вдосконалення інструментів для розробників.

- 27. Аналіз та вдосконалення системи автоматизації тестування.
- 28. Розробка або рефакторинг інструменту аналізу якості коду.
- 29. Оптимізація процесів CI/CD у програмному проєкті.
- 30. Рефакторинг інструменту для роботи з UML-моделлями.
- 31. Розробка або вдосконалення системи моніторингу та логування.

ДОДАТОК Г

Структура змісту КП спеціальності F2 «Інженерія програмного забезпечення»

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ ТА ФОРМУВАННЯ ВИМОГ ДО ВДОСКОНАЛЕННЯ

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВДОСКОНАЛЕНЬ ТА АРХІТЕКТУРИ СИСТЕМИ

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

- специфікації вимог
- лістинги коду
- розширені UML-моделі
- тощо