ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ

«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУТ ІМЕНІ ТАРАСА ШЕВЧЕНКА»

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**

**до виконання та оформлення курсової роботи з дисципліни «Вступ до об‘єктно-орієнтованого програмування»**

Київ 2023

Методичні рекомендації до виконання та оформлення курсової роботи з дисципліни «Вступ до об‘єктно-орієнтованого програмування» для студентів напряму підготовки 050103 «Програмна інженерія» спеціальності 5.05010301 «Розробка програмного забезпечення» — 63 стор.

Розробник Лумпова Тетяна Іванівна, к.е.н.

|  |  |
| --- | --- |
| ЗАТВЕРДЖЕНО | |
| Голова циклової комісії  програмування та інформаційних технологій | |
| ( ) | |
| (підпис) (прізвище та ініціали) | |
| Протокол №1 від «\_\_» \_\_\_\_\_\_ 202\_ року  . | |

Схвалено Методичною радою

ВСП "Оптико-механічний фаховий коледж КНУ імені Тараса Шевченка"

Протокол від «\_\_\_\_\_» \_\_ серпня\_ 2023\_ року № \_\_\_\_\_\_

Голова Методичної ради \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. В. Нестеренко  
 (підпис)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023\_\_\_\_ року

**Зміст**

[**Вступ** 5](#_Toc158061979)

[**1 Загальні положення** 5](#_Toc158061980)

[**2 Опис розділів курсової роботи** 10](#_Toc158061981)

[***2.1 Титульний аркуш*** 10](#_Toc158061982)

[***2.2 Зміст*** 10](#_Toc158061983)

[***2.3. Реферат*** 10](#_Toc158061984)

[***2.4 Перелік скорочень, умовних позначень, символів, одиниць і термінів*** 11](#_Toc158061985)

[***2.5 Вступ*** 11](#_Toc158061986)

[***2.6 Основна частина*** 12](#_Toc158061987)

[***2.6.1 Розділ1. Опис напряму розроблення*** 13](#_Toc158061988)

[***2.6.2 Розділ 2. Опис проектного рішення*** 15](#_Toc158061989)

[***2.7 Висновки*** 26](#_Toc158061990)

[***2.8 Перелік використаних джерел*** 27](#_Toc158061991)

[***2.9 Додатки*** 27](#_Toc158061992)

[**3 Вимоги до оформлення курсової роботи** 28](#_Toc158061993)

[***3.1 Загальні вимоги*** 28](#_Toc158061994)

[***3.2 Оформлення заголовків*** 29](#_Toc158061995)

[***3.3 Нумерація сторінок*** 30](#_Toc158061996)

[***3.4 Нумерація розділів, підрозділів, пунктів, підпунктів*** 30](#_Toc158061997)

[***3.5 Рисунки*** 30](#_Toc158061998)

[***3.6 Таблиці*** 32](#_Toc158061999)

[***3.7 Переліки*** 34](#_Toc158062000)

[***3.8 Примітки*** 34](#_Toc158062001)

[***3.9 Виноски*** 35](#_Toc158062002)

[***3.10 Формули*** 35](#_Toc158062003)

[***3.11 Посилання*** 37](#_Toc158062004)

[**3.12 Додатки** 38](#_Toc158062005)

[**3.13 Оформлення коду** 39](#_Toc158062006)

[**3.13.1. Форматування коду** 39](#_Toc158062007)

[**3.13.2 Угоди про найменування** 41](#_Toc158062008)

[**3.13.3 Загальні принципи розробки** 42](#_Toc158062009)

[**4 Застереження щодо плагіату в курсовому проекті.** 43](#_Toc158062010)

[**5 Захист та оцінювання курсової роботи** 44](#_Toc158062011)

[***5.1 Захист курсової роботи*** 44](#_Toc158062012)

[***5.2 Оцінювання курсової роботи*** 44](#_Toc158062013)

[**5.** **Перелік використаних джерел.** 45](#_Toc158062014)

[**Додаток 1. Календарний план графік** 47](#_Toc158062015)

[**Додаток 2. Зразок титульного аркушу до курсової роботи** 48](#_Toc158062016)

[**Додаток 3. Зразок оформлення змісту** 49](#_Toc158062017)

[**Додаток 4. Зразок оформлення реферату** 50](#_Toc158062018)

[**Додаток 5. Зразок оформлення прийнятих скорочень** 51](#_Toc158062019)

[**Додаток 6. Зразок оформлення використаних джерел** 52](#_Toc158062020)

[**Додаток 7. Критерії оцінювання курсової роботи** 53](#_Toc158062021)

[**Додаток 8. Порядок проведення захисту курсової роботи** 55](#_Toc158062022)

[**Додаток 9. Рекомендації щодо написання доповіді та створення презентації до захисту курсової роботи, підготовки матеріалів у курсовій роботі** 57](#_Toc158062023)

[**Додаток 10. Попередній захис**т 60](#_Toc158062024)

[**Додаток 11. Зразки шаблонів аркушів курсової роботи.** 61](#_Toc158062025)

# **Вступ**

Курсовий проект – це самостійна наукова праця майбутнього програмного інженера. При її виконанні студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, засвоює методику експериментальних досліджень, оволодіває навичками співставлення результатів своїх досліджень з науковими даними, аналізу, узагальнення і літературного оформлення одержаних результатів з теми дослідження, набуває вміння вести науковий пошук, що розвиває в молодого фахівця творчий підхід до роботи.

Курсова робота (далі – КР) з дисципліни «Вступ до об‘єктно-орієнтованого програмування» виконується студентами як підсумкове завдання та відіграє роль підсумкового контролю з дисципліни.

До спеціальних, фахових компетентностей ОПП, що пов’язані з цією курсовою роботою, віднесені:

**СК 1**. Здатність до алгоритмічного та логічного мислення.

**СК 2**. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводу програмного забезпечення та визнання важливості навчання протягом усього життя.

**СК 8**. Здатність розробляти ефективні алгоритми розв‘язання завдань в залежності від предметної області, застосовуючи існуючи типові алгоритми, перетворювати їх на програмний код.

**СК10.**Здатність обгрунтовано обирати та освоювати інструментальні засоби розробки та супроводу програмного засобу.

**СК12.**Здатність розробляти архітектури, модулі та компоненти програмного забезпечення.

# **1 Загальні положення**

Результат роботи студента над курсовим проектом - придбання практичних навичок з програмування та конструювання програмного забезпечення [1], зокрема по:

* аналізу предметної області задачі, визначення вимог до програмного забезпечення (далі – ПЗ), яке реалізується у вигляді консольного застосунку методами об‘єктно-орієнтованого програмування (далі – ООП) ;
* створенню концептуальної та логічної моделі застосунку;
* створенню процесної схеми роботи застосунку як стандартної послідовності виконання робіт (технологічної схеми) і схеми за якою працює застосунок;
* плануванню процесу розроблення ПЗ;
* розробленню застосунку;
* тестування застосунку;
* створенню супровідної документації до ПЗ.

*Мета курсової роботи:*

* глибоке і творче вивчення досліджуваної проблеми на основі опрацювання наукових та навчальних джерел, періодичних видань, законодавчих актів;
* систематизація і розширення теоретичних знань;
* подальший розвиток вміння збирати, аналізувати, систематизувати та використовувати джерела науково-технічної літератури;
* уможливлення виявлення здатності студентів застосовувати отриманні знання при вирішенні практичних завдань, формулювати висновки, пропозиції, рекомендації з предмету дослідження, а також реалізовувати свої ідеї у вигляді програмного продукту;
* закріплення практичних навичок, одержаних студентами під час навчання;
* визначення рівня сформованості вміння використовувати теоретичні знання в практичній діяльності;
* розвинення навичок пояснення та узагальнення результатів емпіричного дослідження, а також виявлення загальних закономірностей.

*Вимоги до курсової роботи:*

* чітка характеристика предмета, мети і методів дослідження предметної області, опис та аналізпроведених практичних дій;
* практична значимість проведених досліджень;
* узагальнення результатів досліджень, обґрунтування висновків та практичних рекомендацій;
* оформлення роботи виконується відповідно до вимог, наданих в [2, 3].

Під час роботи над курсовою студент має

*знати*:

* основні поняття та терміни, що використовуються у сфері програмування;
* особливості проектування програмного продукту та окремих його функціональних модулів;
* функціональні можливості основних модулів (бібліотек) обраної мови програмування;
* основні алгоритми та шаблони ООП.

*вміти:*

* аналізувати, узагальнювати, досліджувати наукові джерела та інформаційні бази даних;
* працювати з науковими статтями, монографіями, а також з методичними, інструктивними матеріалами, критично аналізувати та виявляти їхні позитивні й негативні аспекти;
* використовувати сучасний інструментарій реалізації програмних продуктів, які складаються з оптимізованих алгоритмічних структур об’єктно-орієнтованої парадигми програмування;
* писати програмний код, який відповідає сучасним стандартам іменування об'єктів, коментування та логічного структурування;
* проводити якісне тестування власних програмних розробок;
* використовувати математичні, статистичні, експертні та інші способи оброблення інформації;
* узагальнювати й аналізувати фактичний матеріал, робити відповідні як теоретичні, так і прикладні висновки;
* логічно та чітко викладати матеріал, давати ґрунтовні оцінки різним явищам і процесам;
* швидко знаходити та опановувати інструментарій, призначений для вирішення поточної задачі;
* оформляти результати розроблення програмного продукту відповідно до існуючих стандартів.

КР для розроблення програмного застосунку може виконуватися індивідуально або командою з чітким розділенням повноважень. При командній роботі студенти об‘єднуються в групи по 2-3 чоловіка, обирають тему завдання і визначають провідного виконавця в групі/команді. Відповідно, якщо студенти обирають виконання КР групою, то кількість реалізованих класів та функцій, які повинен надавати застосунок є більшою ніж при індивідуальному виконанні, а також повинен надаватися розподіл робіт по створенню застосунку, які самостійно визначає група студентів разом зі способом роботи у віддаленому режимі (наприклад, створення репозитарію в githab, завантаження програмних артефактів таких як тексти програм, постановки задач тощо) та методом організації співпраці. В пояснювальній записці провідного виконавця відображуються організація процесу спільної співпраці та проблемні питання, які виникали. Повинна бути сформована таблиця розподілу робіт в групі, де визначаються дії кожного студента в групі.

Оскільки КР є самостійною роботою студента, вимоги до функцій, які реалізує застосунок, у студентів буде різним. Так при одноособовому виконанні студентом повинен реалізовуватися в нижче наданий перелік функцій:

* надання переліку функцій для багаторазового обрання потрібної;
* відображення даних про перелік об'єктів у табличному виді;
* введення даних про новий об'єкт в інтерактивному режимі або завантаження з текстового файлу;
* модифікацію даних про попередньо введені об'єкти в інтерактивному режимі;
* видалення попередньо введених об'єктів по одному;
* повне очищення всього переліку об'єктів за однією командою;
* відображення результату роботи всіх методів класів, зокрема, в тестовому режимі;
* сортування переліку даних за кількома найважливішими параметрами;
* пошук запису у переліку за одним з найважливіших параметрів;
* фільтрування даних у переліку за одним з найважливіших параметрів.

При командній роботі додатково повинні бути розроблені такі проектні рішення та функції:

* збереження даних про перелік об'єктів у двійковому форматі;
* читання попередньо збережених даних у двійковому форматі і відображення їх у табличному виді;
* створення таблиці рішень та контроль обрання режиму відповідно до цієї таблиці;
* експорт даних об‘єктів та інформації про перелік об'єктів у текстовий файл;
* імпорт даних об‘єктів з текстового файлу в csv-форматі і відображення їх у табличному виді.

При одноособовому виконанні студент створює не менше 2 зв‘язаних класів, при командній роботі – не менше 4 зв‘язаних класів. Якщо студент або команда вважає недоцільним виконувати всі функції для всіх класів, то в КР надається відповідне обґрунтування, але хоча б до одного класу повинні бути реалізовані всі функції.

Безпосереднє керівництво і контроль за написанням курсової роботи здійснює керівник, відповідно до затвердженого графіка. Про виконану роботу керівник складає відгук, в якому повинні бути висвітлені наступні питання:

* відповідність викладеного матеріалу завданню;
* об'єм виконаної роботи;
* позитивні та негативні сторони виконаної роботи;
* ініціатива та самостійність студента при виконанні роботи;
* вміння користуватись літературою;
* оцінка виконаної роботи.

Пояснювальна записка до курсового проекту, графічні схеми та розроблене програмне забезпечення (далі – ПЗ) надаються керівнику на перевірку відповідно до плану і не пізніше, ніж за 20 днів до захисту, для визначення ступеня готовності проекту. Після перевірки студент усуває наявні недоліки і готує додаткові матеріали для захисту. Попередній захист Курсового проекту може бути передбачений за 20 днів до захисту.

Захист Курсового проекту проводиться перед комісією, до складу якої входять провідні викладачі дисциплін «Програмування», «Вступ до об‘єктно-орієнтованого програмування», «Програмна інженерія» і «Конструювання програмного забезпечення» (не менше двох викладачів циклової комісії спеціальності «Розробка програмного забезпечення», або представників адміністрації технікуму). Захист Курсових проектів відкритий, що передбачає участь як викладачів, так і студентів усіх спеціальностей технікуму.

***Курсова робота є самостійною роботою студента. Відповідальність за правильність аналітичних висновків, результатів розрахунків і моделювання, а також оформлення несе студент - автор роботи****.*

Під час виконання роботи ***потрібно дотримуватися принципів академічної доброчесності*** – забороняється використовувати результати чужої роботи без посилання на автора, це включає рисунки, фрагменти текстів, обчислення, алгоритми та текст програми.

# **2 Опис розділів курсової роботи**

Структурними елементами звіту з виконання КР є:

1. Титульний аркуш (Додаток 2)
2. Зміст (Додаток 3)
3. Реферат (Додаток 4)
4. Перелік скорочень, умовних позначень, символів, одиниць і термінів (Додаток 5)
5. Вступ
6. Основна частина
7. Висновки
8. Перелік використаних джерел інформації (Додаток 6)
9. Додатки (технічний опис, інструкція з експлуатації, код програми, знімки екранних форм і т. п.).

В Додатку 7 цього документу надаються шаблони оформлення аркушу змісту та наступних після нього аркушів.

## ***2.1 Титульний аркуш***

Титульний аркуш (Додаток 2) є першою сторінкою звіту, на якому вказується найменування організації, назва теми роботи, прізвище та ініціали керівника роботи, прізвище та ініціали, група, курс виконавця роботи, місце та дата виконання роботи.

## ***2.2 Зміст***

Зміст (Додаток 3) включає найменування усіх структурних елементів звіту з вказівкою номерів сторінок в курсовій роботі .

## ***2.3. Реферат***

Реферат призначений для ознайомлення з курсовою роботою. Він має бути стислим, інформативним і містити відомості, які дозволяють прийняти рішення про доцільність читання всієї курсової роботи.

Реферат (Додаток 5) повинен бути не менше 0,5 сторінки, але й не більше 1 сторінки і містити:

– дані про кількість сторінок, ілюстрацій, таблиць, використаних джерел, додатків (якщо є),

– перелік ключових слів, записаних великими літерами,

– стислий опис роботи.

У переліку ключових слів зазвичай наводять 5–15 слів або словосполучень, необхідних для розкриття суті роботи. Їх записують за абеткою мови роботи великими літерами в рядок із прямим порядком слів у називному відмінку однини та розділяють комами.

Стислий опис роботи зображує зміст роботи в такій послідовності:

– об’єкт дослідження та розроблення;

– мета роботи;

– методи та інструменти дослідження та розроблення;

– отримані результати та їх значення;

– інформація щодо можливого впровадження;

– рекомендації щодо використання результатів роботи;

– висновки та пропозиції щодо розвитку об’єкта дослідження та розроблення й доцільності продовження досліджень та розробок.

Якщо деякі з наведених відомостей відсутні, то інші викладають, зберігаючи вказану послідовність.

## ***2.4 Перелік скорочень, умовних позначень, символів, одиниць і термінів***

Скорочення, умовні позначення повинні бути подані у вигляді окремого списку, якщо їх в роботі більше 20 та кожне з них зустрічається в роботі більше 3 разів. У протилежному випадку вони розшифровуються безпосередньо у тексті при першому згадуванні. Скорочення, умовні позначення повинні надаватися за абеткою, спочатку україномовні, а потім іноземною мовою.

## ***2.5 Вступ***

У вступі вказується назва КР, формулюється мета роботи й завдання роботи, об’єкт і методи дослідження та розроблення, сфери застосування, оцінка сучасного стану проблеми, що розв’язується. Доцільно визначити актуальність теми КР.

Обсяг вступу – до 1 сторінки.

Мета й завдання роботи мають бути взаємозв’язаними й розкривати тему, заявлену в назві роботи. Мету зазвичай формулюють одним реченням, яке має вказувати, що саме потрібно *встановити, визначити, виявити, з’ясувати, створити*.

Для досягнення поставленої мети зазвичай потрібно розв’язати кілька завдань. Формулювання завдань каже, що саме має бути зроблено: *проаналізувати, визначити особливості, систематизувати існуючий досвід, виокремити, дослідити й описати, розглянути, з’ясувати, простежити, класифікувати, експериментально перевірити й обґрунтувати, спроектувати та розробити, визначити тенденції, окреслити шляхи підвищення ефективності, розробити рекомендації* тощо.

Об’єкт, методи й засоби дослідження або розроблення надаються через характеристику процесів, явищ тощо, обраних для дослідження або автоматизації; опис методів й засобів, за допомогою яких розв’язуються конкретні завдання в роботі. Можливі сфери застосування можуть бути як практичними, так і науковими.

При командній роботі тут також визначають склад команди (з вказівкою провідного члена команди), свій внесок в розроблення застосунку та надають посилання на підрозділ розділу 2 КР, де визначається організація розроблення програмного застосунку.

У вступі й далі в тексті не можна використовувати скорочені слова та терміни, крім загальноприйнятих. Також не бажано вживати іншомовні слова та терміни за наявності рівнозначних слів і термінів у мові, якою написано роботу.

## ***2.6 Основна частина***

Основна частина роботи складається з двох розділів, загальним обсягом 15-30 сторінок (код застосунку надається в додатках і не входить до загального обсягу). Вона повинна містити дані, що відображають суть, методику та результати роботи, а саме:

– розкрити сутність всіх понять, які використовуються і пов’язані з темою дослідження та розроблення;

- проаналізувати і узагальнити різні наукові підходи та точки зору науковців щодо обрання інструментарію розроблення ПЗ;

- подати огляд літератури, яку студентство використовувало при роботі над КР (підручники, стандарти, наукові статті щодо проблематики ООП);

- обґрунтувати власну точку зору щодо використання застосованих методів ООП (наприклад, застосування віртуальних функцій), навести статистичний матеріал з основних питань, подати діаграми, схеми, графіки, алгоритми, тексти програм та дати їм пояснення;

Розділи можуть розбиватися на підрозділи, підрозділи на пункти. Пункти (за потреби) поділяють на підпункти. Заголовки розділів, підрозділів і пунктів мають чітко й стисло відбивати їх зміст. Розділ 1, зазвичай, є теоретичним і розкриває поняття, класифікацію, можливо, історію розвитку тематики. Подальші розділи представляють сутність, методи й засоби розв’язання поставлених задач і результати роботи. Всі розділи та підрозділи потрібно логічно пов'язати між собою.

### ***2.6.1 Розділ1. Опис напряму розроблення***

В Розділі 1 потрібно розкрити такі питання.

1. Постановка задачі.

Потрібно розкрити такі аспекти:

* Мета розробки ПЗ, призначення ПЗ
* Вхідні дані.
* Вихідні дані, очікувані результати.
* Функції обробки інформації (коротка технологічна схема за принципом введення інформації → контроль інформації → збереження/накопичення → оброблення → формування результатів ), які відображують через які процеси вхідні дані трансформуються у вихідні.
* Вимоги до якості (функціональні та нефункціональні вимоги).
* Інші вимоги.

Постановка задачі є відповіддю на серію питань такого роду:

* як визначити рішення;
* які зроблені припущення та т. п.
* якого позитивного результату слід очікувати від автоматизованого розв’язання задачі, що буде покращено, прискорене, оптимізовано, зекономлено при використанні ПЗ;
* яке апаратне забезпечення необхідно для автоматизованого вирішення задачі, яка комп’ютерна платформа необхідна, яка операційна система і яке додаткове програмне забезпечення для проекту;
* який формат вихідних (вхідних) даних, які дані і в якій формі необхідні для розв’язання задачі;
* який формат проміжних і вихідних даних, які дані і в якій формі необхідно отримати;
* які функції обробки інформації повинна забезпечити програма;
* які режими обробки інформації повинна забезпечити програма;
* який інтерфейс користувача програми потрібно забезпечити;
* якою повинна бути «глибина» опрацювання користувацької документації та експлуатаційних документів, керівництв по використанню програми, хто і як буде застосовувати результати роботи програми.

Таким чином, коротко можна сказати, що на етапі постановки задачі необхідно: опис вихідних даних і результату; формалізація завдання; опис поведінки програми в особливих випадках (якщо такі є). Визначаються властивості, якими повинен володіти застосунок в кінцевому вигляді (задум), в загальних рисах описуються функції застосунку, характеристики інтерфейсу.

Ці питання доцільно описати коротко, оскільки детально це розкривається в розділі 2. Кожна визначена позиція повинна розкриватися у розділі 2.

1. Опис предметної області об‘єкту, для якого має виконуватися розроблення ПЗ. Наприклад, причини, які викликали потребу в автоматизації певних сфер діяльності, застосуванні нових інформаційних технологій або модернізації існуючих.
2. Огляд (або систематизація) і оцінка наукової літератури з питань розроблення ПЗ для обраної предметної області з погляду можливостей щодо реалізації програмного продукту для розглянутого об‘єкту.

Надається короткий огляд наукових джерел з теми курсової, з якими потрібно ознайомитися для якісного вирішення задачі. Всі джерела вносяться до переліку використаних джерел, а тут надаються посилання. Огляд наукової літератури є зведеною характеристикою питань, що розглядаються та надаються в переліку джерел, без їх критичного оцінювання. Його завдання – адекватне відображення змісту джерел, хоча це не виключає виокремлення найважливіших і найактуальніших документів.

1. Огляд існуючих програмних засобів – аналогів для розроблення програмного застосунку.

Надається короткий опис, аналіз їх переваг і недоліків та обґрунтування власного підходу до обрання програмних засобів розробки. Після аналізу цього матеріалу визначається, чому запропоноване інструментальне ПЗ доцільно використовувати з наявного на ринку (наприклад, дешевизна, можливість розгорнути на апаратній платформі розробника без додаткової закупівлі техніки та системного ПЗ).

В цілому в Розділі 1 студент демонструє своє вміння аналізувати вимоги замовника до програмного продукту, формулювати положення технічного завдання та постановки задачі, аналізувати предметну область з метою проведення перед проектного дослідження.

### ***2.6.2 Розділ 2. Опис проектного рішення***

В цьому розділі надаються вимоги до ПЗ з погляду розробника та розроблені відповідно до цих вимог проектні рішення, які, в свою чергу, включають абстрактні моделі, створені в математичних термінах і візуальних нотаціях.

В цьому розділі можна виділити декілька відносно самостійних підрозділів, що розкривають теоретичні та практичні аспекти розробки ПЗ. З’ясовується семантика класів і об’єктів — визначається поведінку і атрибути кожної абстракції, визначається структура вхідних та вихідних даних, структура функціональних модулів та взаємозв’язків між ними, розробляється інтерфейс користувача. Розділ повинен бути максимально насичений фактичною інформацією (таблиці, схеми, UML діаграми) та всебічно відображати аспекти діяльності розробленого ПЗ.

В розділі розкриваються такі питання як окремі підрозділи (курсивом виділено назву підрозділу в КР): алгоритм рішення, модель програмного рішення, ієрархічно-функціональна схема, методи та механізми ООП, використані в застосунку, опис програмного інтерфейсу, тестування програмного застосунку і результати його виконання,коротка інструкція користувача, організація розроблення програмного застосунку, .

1. *Алгоритм рішення* (укрупнена схема алгоритму рішення), технологія оброблення інформації.

На основі постановки задачі формується опис алгоритму роботи програми, надається блок-схема алгоритму і детальний опис роботи всіх блоків алгоритму. До блок – схеми доцільно надати схему технологічного процесу обробки інформації як сукупності взаємозв’язаних технологічних операцій, які виконуються над інформацією у певній послідовності. Технологічна операція — це комплекс дій з інформацією та її носіями, які виконуються на одному робочому місці. Приклад простої блок-схеми надано на рис.1, приклад простої схеми технологічного процесу обробки інформації надано на рис 2.

В описі алгоритму визначаються особливості виконання окремих технологічних операцій (введення, оброблення даних, формування вихідних даних тощо), надаються коментарі щодо нештатних ситуацій (наприклад, введення неправильних даних, помилки введення-виведення, порушення послідовності виконання операцій).

При описі алгоритму роботи програмного додатку, потрібно зазначити основні зв’язки, за якими буде функціонувати додаток, послідовність виконання його функцій, тощо.

При виконанні КР командою визначаються і виділяються на схемах блоки операцій, які реалізуються студентом – членом команди.

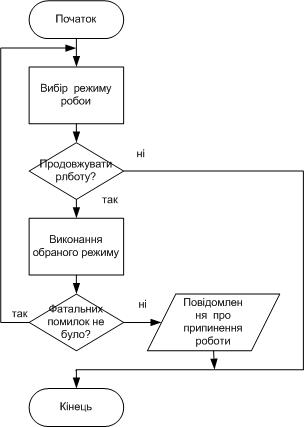


Рисунок – 1 Приклад простої блок-схеми

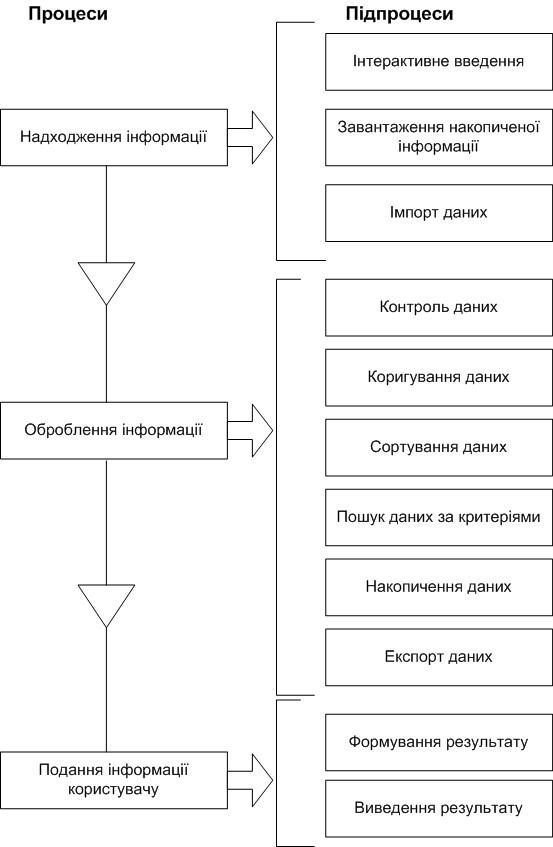


Рисунок – 2 Приклад простої схеми технологічного процесу обробки інформації

1. *Модель програмного рішення*.

На основі інформації, закладеної у підрозділі "Алгоритм рішення", має бути побудована UML-діаграма класів.

UML — уніфікована мова моделювання, яка використовується у парадигмі об'єктно-орієнтованого програмування. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення, використовує графічні позначення для створення абстрактної моделі системи. В такій діаграмі повинні знайти відображення всі структурні об'єкти програмного додатку, а також їх важливі методи та властивості, які відповідають за взаємодію класів між собою та безпосереднє виконання поставленої задачі.

В діаграмі класів визначаються перелік та опис призначення об'єктів, а також структури об'єктів (поля і методи). Приклад діаграми класів надається на рис.3. При виконанні КР командою на UML-діаграмі класів визначаються і виділяються класи, які реалізуються студентом – членом команди.

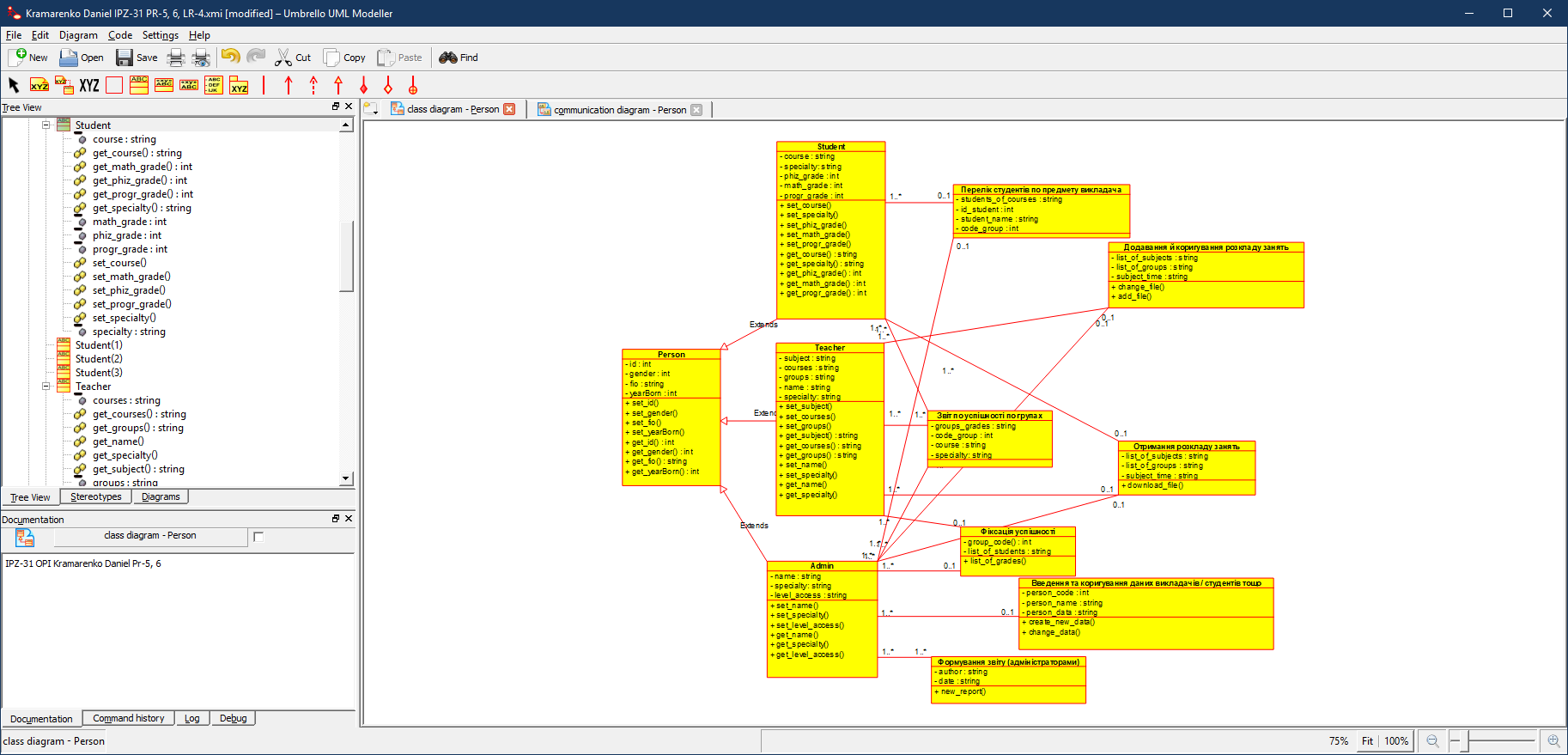


Рисунок – 3 Приклад UML-діаграми

До кожного класу надається в табличному вигляді перелік методів з наданням опису призначення методу та поясненнями щодо призначення параметрів методів.

1. *Ієрархічно-функціональна схема застосунку*

Ієрархічно-функціональна схема є результатом функціональної декомпозиції і надаєієрархічну схему модулів/ функцій з описом їх призначення. Схема представляється у вигляді блокової діаграми, де блоки відповідають основним модулям/функціям, а існуючі зв'язки між ними позначаються лініями, можливо зі стрілками. В цій схемі доцільно визначити всі компоненти програми (функції, класи). Надаючи ідентифікатори компонентів, після схеми надається таблиця, де для кожного компонента надається його призначення та виконувані функції. Приклад простої ієрархічно-функціональної схеми застосунку надається на рис.4. При виконанні КР командою визначаються і виділяються на схемі модулі/функції, які реалізуються студентом – членом команди.

****

Рисунок – 4 Приклад простої ієрархічно-функціональної схеми застосунку

1. *Методи та механізми ООП, використані в застосунку*

В цьому підрозділі наводяться основні відомості про об'єктно-орієнтований підхід до проектування програмного забезпечення, а також методи та механізми ООП, використані в застосунку, і обґрунтування їх використання. До таких методів та механізмів відносяться:

* різні типи конструкторів;
* виокремлення опису класу;
* статичні та константні елементи класів;
* відношення клас-підклас;
* використання перевантажених функцій;
* вбудовані класи;
* делегування конструкторів;
* визначення відношень між класами (залежність, асоціація, узагальнення);
* дружні функції та класи;
* обрання специфікаторів доступу, спеціальних методів при спадкуванні;
* перевантаження функцій та операцій;
* поліморфізм та віртуальні функції;
* перевантаження та перевизначення функцій;
* віртуальні функції;
* абстрактні класи;
* класи-інтерфейси;
* ієрархія класів;
* множинне наслідування;
* класи потоків введення/виведення;
* перевантаження операторів потокового введення/виведення;
* обробка виключень та виняткових ситуацій;
* шаблони функцій та класів;
* класи колекцій стандартної бібліотеки.

До використаних методів та механізмів надається коротке пояснення щодо їх використання в застосунку та досягнених позитивних результатів.

В цьому підрозділі також визначають класи стандартної бібліотеки С++, використані в програмі, надаючи їх переліком с визначенням призначення.

1. *Опис програмного інтерфейсу*.

Зазвичай спілкування користувача з консольним застосунком здійснюється через меню, де користувачу надається перелік функцій, які надає застосунок. Таке меню може бути ієрархічне, багаторівневе; назва функції повинна бути змістовною і лаконічно.. В КР надається схема меню консольного застосунку (приклад меню див. рис.5) і короткі пояснення до кожної гілки меню. При виконанні КР командою визначаються і виділяються на схемі модулі/функції, які реалізуються студентом – членом команди.

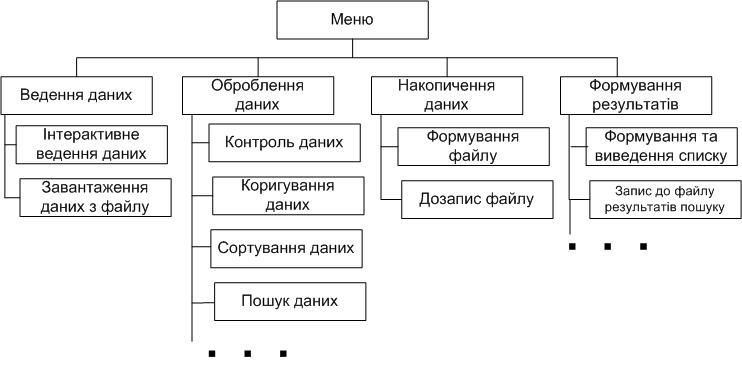


Рисунок – 5 Приклад ієрархічного меню застосунку

Якщо у студентів виникають проблеми з локалізацією текстової інформації, дозволяється надавати меню та повідомлення англійською мовою (не транслітерацією!).

Опис файлів даних та інтерфейсу програми передбачає що, якщо програма використовує файлові об’єкти як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то в даному підрозділі слід навести опис формату цих файлів. Він може бути виконаний у текстовому, табличному чи графічному вигляді. Можна додавати в додатку до нього приклади реальних файлів із даними програми.

1. *Тестування програмного застосунку і результати її виконання*

Тестування програмного забезпечення (Software Testing)– це перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраних певним чином.

Існує два основних різновиду тестування: функціональне та модульне. Функціональне тестування перевіряє, як програма виконує свої функції в умовах коректних та некоректних дій користувача. Воно планується на етапі проектування и виконується у міру того, як кодуються окремі функції програми і перед захистом КР.

Модульне тестування - це тестування програми на рівні окремо взятих модулів, функцій або класів. Мета модульного тестування полягає у виявленні локалізованих в модулі помилок у реалізації алгоритмів, а також у визначенні ступеня готовності системи до переходу на наступний рівень розробки і тестування. Модульне тестування проводиться за принципом "білого ящика". На рівні модульного тестування найпростіше виявити дефекти, пов'язані з алгоритмічними помилками і помилками кодування алгоритмів. Помилки, пов'язані з невірним трактуванням даних, некоректною реалізацією інтерфейсів, сумісністю, продуктивністю тощо, зазвичай виявляються на більш пізніх стадіях тестування. Модульне тестування зазвичай передбачає створення навколо кожного модуля певного середовища. Модульне тестування робиться одночасно с кодуванням и має за мету автоматичну перевірку окремих частин програми. Виконання модульного тестування є бажаним, але не є обов’язковим.

В цьому підрозділі описується процес проведення модульного тестування, який відбувався під час розроблення. Може надаватися опис тестових даних, які використовувались для перевірки працездатності програми, сам набір тестових даних може надаватися в додатку. До набору повинні надаватися коментарі щодо очікуваного виявлення помилок. Необхідно надати різні тести, які передбачають різні послідовності взаємодії з програмним забезпеченням для тестування. Варіанти тестування й отримані результати повинні бути формально зафіксовані.

У даному підрозділі також потрібно описати методику демонстраційного тестування програми, а відповідні тестові дані та результати роботи програми надаються в додатках. Демонстраційне тестування призначене показати працездатність застосунку на тестових даних при виконанні тестового сценарію. Тестові дані – це вхідні і вихідні дані, підібрані так, щоб для заданих вхідних даних були відомі вихідні. Для вхідних даних програма (модуль), що перевіряється, видає свої вихідні дані. Якщо результат роботи програми не збігається з тестовими  даними, то це значить, що в програмі є помилка. Якщо тестові дані містять помилку, то результат демонстраційного тестування демонструє дії програми по обробленню такої ситуації (наприклад, виведення повідомлення про помилкові дані, виконання дій передбачених алгоритмом в цій ситуації).

Типова процедура тестування полягає у підготовці та виконанні тестових прикладів (або просто тестів). Кожен тестовий приклад перевіряє одну "ситуацію" в поведінці модуля і складається зі списку значень, переданих на вхід модуля, описи запуску і виконання переробки даних - тестового сценарію і списку значень, які очікуються на виході модуля в разі його коректної поведінки. Виконання починається з передачі модулю вхідних даних і запуску сценарію. Реальні вихідні дані,отримані від модуля в результаті виконання сценарію, зберігаються і порівнюються з очікуваними. У разі їх збігу тест вважається пройденим, в іншому випадку - НЕ пройденим. Кожний не пройдений тест вказує на дефект або в тестуємому модулі або в описі тесту.

Сукупність описів тестових прикладів становить тест-план, який визначає процедуру тестування програмного модуля. Тест-план задає не тільки самі тестові приклади, а й порядок їх слідування, який також може бути важливий. Приклад тест – плану наданий в табл.1.

Таблиця 1 – Приклад тест - плану

| Визначення мети | Послідовність дій | Очікуваний результат |
| --- | --- | --- |
| Перевірка інтерактивноговведення даних та запису їх до файлу | 1. Обрати функцію "Інтерактивневведення даних"  2. Обрати клас, до якого вводяться дані (наприклад, "Студент")  3. Ввести дані по студенту.  4. На запит "Дозаписати дані до файлу?" відповісти "Так"  5. На запит "Введення завершено" відповісти "Так". | Повернення відбувається в головне меню. При перегляді даних у файлі введена інформація збережена правильно. |
| Перевірка реакції на неправильне обрання позиції меню. | Першою обирається позиція меню "Сортування даних" | Видається повідомлення "Введіть /завантажте дані", відбувається повернення в головне меню. |

У підрозділі потрібно зробити висновок, який підтверджує (або заперечує) працездатність програми.

1. Керівництво для користувача програми

Наводиться інструкція по роботі з програмою, в якій описується склад програмного продукту (імена всіх файлів та каталогів, з яких складається програма із зазначенням їх розміру і призначення, розташування на диску), інтерфейс (зовнішній вигляд головного меню, вікна з додатковими меню), надається послідовність дій по реалізації основних функцій, а також перелік повідомлень, які видає програма користувачу, з визначенням причинам видачі повідомлення та дії, які очікуються від користувача для успішного продовження роботи. При використанні вхідних даних вказуються характер, організація, попередня підготовка, формат, опис і засіб кодування вхідних даних. Описуються характер і організація вихідних даних.

В цьому підрозділі надаються короткі відомості про, а сам документ розташовують в додатку. Саме Керівництво доцільно писати лаконічно, можна надавати посилання на основний текст КР.

В цьому підрозділі також надається специфікація до ПЗ як перелік артефактів, необхідних для приймальних випробувань (в даному випадку – захисту курсової) та експлуатації.

1. *Організація розроблення програмного застосунку*

Визначаються етапи розроблення ПЗ та надаються у вигляді загального календарного графіку розробки ПЗ і деталізованого плану. При роботі команди фіксуються конкретні виконавці.

Розроблення програми при об'єктно-орієнтованому підході розбивається на розроблення системи класів та глобальних типів даних, реалізацію сценаріїв використання, методів, дружних та глобальних функцій, створення об'єктів та організацію взаємодії між ними в головній програмі для досягнення поставленої мети.

При розробленні календарного графіку потрібно враховувати технологію розроблення програм за об‘єктно-орієнтованим підходом.

Об'єктно-орієнтований підхід як ключовий метод розроблення складних програмних проектів, завдяки принципам інкапсуляції, ієрархії класів, успадкуванню та поліморфізму дає можливість виділити всі спільні та відмінні риси окремих елементів поставленого завдання і записати їх у програмі в найлаконічнішій формі.

Розроблення програми при об'єктно-орієнтованому підході розбивається на розроблення системи класів та глобальних типів даних, реалізацію сценаріїв використання, методів, дружних та глобальних функцій, створення об'єктів та організацію взаємодії між ними в головній програмі для досягнення поставленої мети. Всі ці етапи повинні фіксуватися в плані.

В загальних рисах процес, який передує кодуванню та налагодженню, зазвичай має такий перебіг.

Після розроблення постановки задачі та визначення в загальних рисах алгоритму рішення, виконується декомпозиція програми (проектується її системна архітектура) та розробляються основні та альтернативні сценарії використання, визначаються основні об’єкти, методи та алґоритми їх взаємодії.

Дані та дії над ними утворюють об'єкт. Можна уявити програму як набір об'єктів (реально існуючих фізичних об'єктів або абстрактних понять), які взаємодіють між собою.

Наступним етапом є поділ об'єктів на групи, що мають щось спільне (дані, властивості, функціональність). У більшості випадків об'єкти будуть більше чи менше подібні. Деколи в групах можна буде виділити підгрупи тощо та сформувати складне ієрархічне дерево. У кожній групі виділяються найзагальніші властивості, притаманні всім без винятку об'єктам у групі та дії (операції) над ними. Таким чином отримуємо базовий клас (суперклас) для певної групи об'єктів, даними якого будуть загальні параметри об'єктів групи, а методами — дії, які можна проводити над даними будь-якого об'єкта групи.

Далі для кожної з підгруп (у межах виділеного класу об'єктів) додаємо притаманні лише їй властивості, як поля нового породженого від базового похідного класу.

Додаткові дії, властиві кожній підгрупі, стають методами похідних (породжених) класів, а параметри, яких не мали об'єкти базового класу, — даними похідних класів. Породження похідних класів продовжується до тих пір, поки не будуть описані всі параметри об'єктів та дії, які необхідно над ними здійснювати для розв'язання поставлених задач. Описана послідовність повторюється для кожної групи об'єктів у програмі.

Кожна спільна для кількох об'єктів дія (метод, повідомлення, операція) та спільні елементи даних описуються лише один раз у базовому чи похідному класі, більше ніде їх не потрібно повторно описувати.

Якщо деякі дії передбачають виконання операцій над кількома об'єктами (обмін повідомленнями між кількома об'єктами) і при цьому необхідно мати безпосередній доступ до закритих даних класу, то такі дії зазвичай виконуються через дружні функції (методи) для того класу, над даними якого вони працюють. Методи, визначені в базових класах і які будуть використовуватися для перезавантаження в похідних класах, доцільно визначати як віртуальні.

Деякі типи даних, які недоцільно робити класом, оскільки вони не мають власної функціональності, але які неодноразово використовуються в програмі (наприклад, масиви фіксованої довжини, списки чи структури) можна описати, використавши typedef, і помістити в заголовок одного з файлів.

Глобальні методи, які не є методами ні одного з класів (їх можна назвати алґоритмами), також поміщаються в один із \*.срр - файлів, а їх інтерфейсна частина описується разом з описами класів у заголовному файлі (\*.h - файлі).

Описані етапи аналізу стосуються інтерфейсної частини застосунку, їх доцільно зафіксувати в плані, якщо вони виконуються.

Після розроблення інтерфейсної частини розробляють реалізацію методів, дружних та глобальних функцій (методів) для забезпечення виконання сценаріїв використання. Це завдання буде досить простим, якщо на попередньому етапі було правильно спроектовано систему класів і детально спроектовано та розроблено інтерфейсну частину застосунку. Реалізація будь-якого методу розміщується в \*.срр -частині бібліотеки, на відміну від інтерфейсної, розміщеної у \*.h-файлі. Доцільно користуватися вже готовими бібліотечними функціями, шаблонними класами та методами, зокрема STL-засобами замість того, щоб увесь час змінювати дані за допомогою одних і тих же послідовностей операцій. В останньому випадку потрібно зафіксувати на етапі аналізу дії по дослідженню існуючих стандартних засобів, а в підрозділі "*Методи та механізми ООП, використані в застосунку*" визначити використані готові бібліотечні функції, шаблонні класи та методи.

Методи можуть бути реалізованими з використанням поліморфізму та успадкування, використовуючи уже реалізовані методи базових класів. У підрозділі "*Методи та механізми ООП, використані в застосунку*" потрібно описати принцип дії основних методів та відмінності поліморфних (віртуальних) методів від методів базових класів.

Створення об'єктів і розроблення головної програми – етап, де реалізується процес створення самих об'єктів у програмі, функціональність і дані яких розроблені на попередніх етапах. Якщо кількість об'єктів, що будуть створені, наперед невідома, або обсяг пам'яті, яку вони займають, є досить значним, тоді пам'ять для об'єктів слід виділяти "динамічно" з використанням оператора new. У цьому випадку доброю практикою є постійний контроль обсягу вільної пам'яті та використання певних застережних дій (наприклад, видача попереджувальних повідомлень), про що слід обумовити в альтернативних сценаріях. Ці аспекти потрібно зафіксувати в підрозділі "*Методи та механізми ООП, використані в застосунку*".

Створивши об'єкти, можемо реалізувати певну послідовність дій або схему взаємодії між об'єктами, яка призводить до вирішення поставленого завдання. Опис цієї послідовності дій чи схеми взаємодії, можливо, з прикладами для реальних вхідних даних чи повідомлень також може бути зробленим окремим етапом. Такий опис повинен бути достатнім для розуміння всіх деталей реалізації алґоритму та можливих випадків, що зустрічаються в роботі програми.

Для роботи з великою кількістю об’єктів, що займають значний обсяг пам’яті, де передбачені операції копіювання, сортування, переміщення, доцільно з метою економії пам’яті вводити об’єкти вказівників і допоміжні функціональні об’єкти (шаблонні класи порівняння) та інші STL-засоби (контейнери, алґоритми, методи та ітератори). Ці дії також можна виділити окремим етапом, а самі засоби зафіксувати в підрозділі "*Методи та механізми ООП, використані в застосунку*".

Для забезпечення ефективності потокового вводу/виводу потрібно використовувати потокові класи та об’єкти (у тому числі файлові). Аналітичну роботу над цим питанням також доцільно відобразити в плані.

## ***2.7 Висновки***

Основне призначення даного розділу – визначення, в якій мірі результати виконаної роботи відповідають початковим вимогам до програмного продукту. Цей аналіз може бути використаний для формування на його основі планів і перспективних напрямів розвитку продукту та його використання. Виконання аналізу може бути необхідним, щоб показати як можна розширити та доповнити новими функціями розроблений застосунок, або поліпшити якість виконання існуючих функцій.

## ***2.8 Перелік використаних джерел***

Перелік використаних джерел є описом *і*нформаційної бази курсового проекту, яка складається з наукових джерел (монографічні праці, наукові статті, доповіді вітчизняних і зарубіжних вчених з питань дослідження; матеріали науково-практичних конференцій і семінарів); офіційних документів (нормативні документи України та інших країн світу, офіційні матеріали органів законодавчої та виконавчої влади); статистичних джерел (фінансова, податкова, управлінська й статистична звітність суб’єкта господарювання); Інтернет - ресурсів.

Перелік вказує на джерела, використані й зазначені в роботі. У переліку джерел посилання бібліографічні описи подають у порядку, за яким джерела вперше згадано в тексті або за абеткою. Порядкові номери бібліографічних описів у переліку джерел мають відповідати посиланням на них у тексті (номерні посилання). Номери записуються арабськими цифрами з крапкою.

Основні вимоги до переліку джерел:

* відповідність темі курсової роботи;
* різноманітність видів видань (нормативні, навчальні, наукові та ін.);
* на неопубліковані роботи посилання не допускаються;
* бібліографічний описи джерел у переліку оформлюють згідно з діючим стандартом для бібліографічної та видавничої справи [3, 8] (приклади наведено в Додатку 6).

## ***2.9 Додатки***

У додатки включають матеріали, що не ввійшли до основної частини звіту. Зокрема, у додаток можна включити тексти програми, таблиці тощо. Тексти програм повинні починатися з найменування програмних модулів (класів, файлів). Кожен з них повинен починатися з нової сторінки. Кожен клас повинен мати початковий коментар, що описує призначення класу, його взаємодію з іншими класами та інші додаткові відомості, що необхідні для розуміння роботи класу. При написанні програм бажано враховувати необхідність друку кодів програмних модулів та, за можливістю, не припускати розривання рядків коду засобами друку текстів.

Обов‘язковим є надання у додатках текстів програм програмного застосунку, посібника користувача, результати виконання програмного застосунку на тестових даних із використанням знімків екранних форм – "скріншотів".

Кожен додаток починається з нової сторінки. У правому верхньому куті слід написати слово ДОДАТОК. Кожен додаток повинен мати назву. Якщо в роботі 2 чи більше додатків, то їх нумерують послідовно ДОДАТОК 1, ДОДАТОК 2, …

**Обов‘язковими** є додатки:

Тексти програми;

Результати виконання програмного застосунку (надаються скріншоти виконання);

Керівництво для користувача програми.

# **3 Вимоги до оформлення курсової роботи**

Оформлення КР виконується відповідно до вимог, визначених в [1] і які стосовно текстової частини практично повністю наводяться нижче.

## ***3.1 Загальні вимоги***

Роботу оформлюють з урахуванням вимог державних стандартів, описаних нижче. Роботу друкують на одній стороні аркуша білого паперу формату А4 (210х297 мм). Можливий формат А3 (297х420 мм), коли це необхідно для ілюстрацій. Використовують шрифти з мінімальною висотою шрифту 1,8 мм (зазвичай, Times New Roman Cyr, розмір 14 пунктів). Міжрядковий інтервал 1,5, ширина берегів: верхній і нижній – не менше ніж 20 мм, лівий – не менше ніж 25 мм, правий – не менше ніж 10 мм. Абзацний відступ має бути однаковий упродовж усього тексту роботи й дорівнювати 1,27 см.

Окремі слова, формули, знаки можна вписувати чорним чорнилом, тушшю чи пастою. Насиченість знаків вписаного тексту має бути наближеною до насиченості знаків надрукованого тексту. Помилки й графічні неточності можна виправляти на паперовому носії, підчистивши або зафарбувавши білою фарбою і потім вписавши або вдрукувавши на цьому місці правку між рядками чи на рисунках чорним чорнилом, тушшю чи пастою.

Прізвища, назви установ, організацій, фірм та інші власні назви наводять мовою оригіналу. Дозволено транслітерувати власні назви в перекладі на мову роботи, додаючи оригінальну назву в дужках у місці першого згадування в тексті.

У тексті, але не в заголовках, можна скорочувати слова та словосполучення згідно з правописними нормами та ДСТУ 3582.

Позначення фізичних та інших величин мають відповідати установленим у стандартах, зокрема, треба користуватися основними, похідними чи позасистемними одиницями фізичних величин Міжнародної системи одиниць (SI) згідно з ДСТУ 3651.0, ДСТУ 3651.1 і ДСТУ 3651.2.

Числа з розмірністю записують цифрами, а без розмірності – словами («відстань – 8 мм», «відміряти два рази»). Позначення одиниць вимірювання пишуть після числа через пробіл і без перенесення в наступний рядок. Якщо наводиться ряд числових значень однієї фізичної величини, то одиницю її вимірювання вказують після останнього значення («1,5; 1,75; 2 мм»). Позначення величин з граничними відхиленнями записують так: «100 ± 5 мм».

Порядкові числівники записують цифрами з відмінковими закінченнями («9-й день», «4-а лінія»). При кількох порядкових числівниках відмінкове закінчення записують після останнього («3, 4, 5-й графіки»). Кількісні числівники записують без відмінкових закінчень («на 20 аркушах»), не пишуть закінчення в датах («21 жовтня») та при римських числах («XXI століття»).

У тексті не допускаються:

– скорочення слів, крім загальноприйнятих в українській мові;

– професійні або місцеві слова та вирази (техніцизми);

– з'єднання тексту з умовним позначенням фізичних величин за допомогою математичних знаків (слід писати не «швидкість = 5 км/год», а «швидкість дорівнює 5 км/год», не «температура дорівнює –5°С», а «температура дорівнює мінус 5°С»);

– математичні знаки <, >, 0, №, %, sin, cos, tg, log тощо без цифрових або буквених позначень (у тексті слід писати словами «менше», «нуль», «номер», «логарифм» тощо).

## ***3.2 Оформлення заголовків***

Кожна структурна частина починається з нової сторінки. Назви структурних частин друкують у вигляді заголовків **РЕФЕРАТ**, **ЗМІСТ**, **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ**, **ВСТУП**, **РОЗДІЛ …**, **ВИСНОВКИ**, **ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**, **ДОДАТКИ** великими літерами жирним шрифтом симетрично відносно тексту. Заголовки підрозділів та пунктів починають з абзацного відступу і друкують жирним шрифтом малими літерами (окрім першої великої). Усі заголовки не мають підкреслень і крапок у кінці. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою. Переноси частин слів в заголовках не допускаються, слова переносяться цілком.

Відстань між заголовком і текстом – два рядки. Відстань між рядками заголовка така, як у тексті, між заголовком і підзаголовком, а також між підзаголовком і текстом – один рядок.

Не дозволено розміщувати заголовки в останньому рядку сторінки або в її нижній частині, якщо після заголовка є тільки один рядок тексту.

## ***3.3 Нумерація сторінок***

Титульний аркуш має номер 1 у загальній нумерації сторінок, але номер на ньому не ставиться. На інших сторінках порядковий номер вказують в верхньому правому куті. Сторінки нумерують наскрізно арабськими цифрами, включно з додатками.

## ***3.4 Нумерація розділів, підрозділів, пунктів, підпунктів***

Заголовки розділів, підрозділів, пунктів, підпунктів нумерують арабськими цифрами. Розділи нумерують у межах роботи, починаючи з «1». Номер підрозділу включає номер розділу та порядковий номер підрозділу, розділені крапкою. Підрозділи нумерують у межах кожного розділу окремо. В кінці номера крапку не ставлять, а пропускають один знак. Пункти, підпункти нумерують аналогічно. Цифри, які вказують номер, не повинні виступати за абзац.

**Приклади**

**1 ЗАГОЛОВОК РОЗДІЛУ**

**1.1 Заголовок першого підрозділу в розділі та**

**його продовження**

**1.1.1 Заголовок першого пункту підрозділу**

**1.1.1.1 Заголовок першого підпункту першого пункту**

## ***3.5 Рисунки***

Для наочності рекомендується додавати в текст ілюстративні графічні матеріали (ескізи, діаграми, графіки, схеми, фотографії, рисунки, кресленики тощо), які в технічній документації називають рисунками.

Під кожним рисунком має бути підпис, розміщений симетрично до тексту. Підпис починається словом «Рисунок», містить номер і назву, наприклад, «Рисунок 2 – Другий рисунок». Назву записують після тире з великої літери. Крапку в кінці не ставлять, знак переносу не використовують. Назва рисунка має відображати його зміст, бути конкретною та стислою. Якщо з тексту зрозуміло зміст рисунка, його назву можна не наводити.

Рисунки нумерують наскрізно арабськими цифрами, крім рисунків у додатках. Можна нумерувати рисунки в межах кожного розділу. Тоді номер рисунка складається з номера розділу та порядкового номера рисунка в цьому розділі, відокремлених крапкою, наприклад, «Рисунок 2.3 – Третій рисунок другого розділу». Номер рисунка в додатку складається з познаки додатка та порядкового номера рисунка в додатку, відокремлених крапкою, наприклад, «Рисунок А.1 – Перший рисунок додатка А».

Рисунок розміщують після абзаца, де вперше посилаються на нього, або якнайближче до нього на наступній сторінці, а за потреби – в додатках. У тексті має бути посилання на рисунок з його номером, наприклад «…º(рисунок 3.1)» або «… на рис. 3.2». Між рисунком і текстом пропускають один рядок. Рисунок розміщують симетрично до тексту без повороту або з поворотом на кут 90° за годинниковою стрілкою.

Якщо рисунок складається з частин, їх позначають малими буквами українського алфавіту з дужкою, наприклад, «*а*)», «*б*)» під відповідною частиною. Тоді після назви рисунка й двокрапки наводять назву кожної частини за такою формою: «Рисунок 2.1 – Назва рисунка: а – назва першої частини; б – назва другої частини».

Якщо частини рисунка не вміщуються на одній сторінці, то їх переносять на наступні сторінки. Під початком рисунка вказують його повне позначення, а під його продовженнями записують «Рисунок 2.1 (продовження)». Пояснювальні дані розміщують під кожною частиною рисунка.

Якщо в тексті є посилання на складові частини графічного зображення, то на рисунку вказують їх порядкові номери в межах рисунка. Посилання на нумерований елемент рисунка в тексті дається без дужок, наприклад, «кран 2 на рис. 3.1». Розшифровку номерів або літерних позначень та інші пояснювальні дані до рисунка наводять безпосередньо після графічного зображення перед назвою рисунка.

Якщо рисунок є фрагментом повної розробленої схеми, то для всіх компонентів вказують ті позиційні позначення, які вказані на схемі.

## ***3.6 Таблиці***

Таблиця складається з номера таблиці, її назви й самої таблиці відповідно до такої форми.

Рядок із словом «Таблиця», номером та назвою записують з абзацним відступом. Крапку в кінці не ставлять. Якщо назва таблиці довга, то продовжують у наступному рядку під словом «Таблиця». Назва таблиці повинна відбивати її зміст, бути точною й стислою. Якщо з тексту зрозуміло зміст таблиці, її назву можна не наводити.

Таблиця 2.2 – Друга таблиця в розділі 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Головка* → |  | |  | |
|  |  |  | *Заголовок колонки*  *Підзаголовки (за потреби)* |
|  |  |  |  |  |
|  |  |  |  |  |

*Боковик (за потреби ↑*

*заголовки рядків)*

Таблицю розміщують симетрично до тексту безпосередньо після абзацу, в якому її згадано вперше, або на наступній сторінці. Таблицю заповнюють так, щоб було зручно розглядати її без повороту, або з поворотом на кут 90° за годинниковою стрілкою.

Таблиці нумерують аналогічно рисункам (див. підрозділ 3.5). На кожну таблицю має бути посилання в тексті з вказівкою її номера, наприклад, «таблиця 2.2» або «табл. 2.1».

Таблицю утворюють графи (колонки) та рядки. Згори розміщують головку таблиці, в якій вказують заголовки граф. Діагональне ділення головки таблиці не допускається. Ліва графа (боковик) за потреби може містити назви рядків.

Горизонтальні та вертикальні лінії, що розмежовують рядки та колонки таблиці, можна не наводити, якщо це не ускладнює сприйняття таблиці. Висота рядків таблиці повинна бути не менше 8 мм.

Заголовки рядків записують у боковику таблиці в називному відмінку однини, малими буквами, починаючи з великої. Заголовки колонок записують паралельно рядкам таблиці, але за необхідності допускається їх перпендикулярне розташування.

Заголовки починають з великої літери, підзаголовки – з малої літери, якщо вони становлять одне речення із заголовком, або з великої, якщо вони мають самостійне значення. В кінці заголовків і підзаголовків крапки не ставлять. Заголовки й підзаголовки, складені з одного слова, вказують в однині називного відмінку. Можна записувати заголовки та підзаголовки граф таблиці через один інтервал.

Позначення одиниць виміру фізичних величин вказують у заголовках після коми. Якщо всі показники таблиці мають однакову розмірність, то вони виносяться в заголовок таблиці, інакше одиниці виміру наводяться окремо в заголовках колонок або заголовках рядків. Одиниці виміру вказуються в скороченому вигляді.

Для опису інтервалів значень у заголовках граф і рядків можна використовувати слова «більше», «менше», «не більше», «не менше», «у межах». Ці слова розміщують після одиниці фізичної величини, наприклад «Напруга, В, не більше». Також використовують слова «від», «більше» і «до»: «Від 10 до 15», «більше 15», «до 20».

Числа записують посередині графи так. Інтервал вказують від меншого числа до більшого з тире між ними: 12 – 35, 122 – 450. Дробові числа наводять у вигляді десяткових дробів, з однаковою кількістю знаків після коми в одній графі. Розміри в дюймах можна записувати у вигляді 1/2", 3/4", 5/8" тощо *.*

Графу «*№ п/п*» у таблицю не включають. Нумерація колонок або рядків таблиці арабськими цифрами допускається тільки тоді, коли в тексті документа є посилання на них, або при розподілі таблиці на частини, або при перенесенні частини таблиці на наступну сторінку. За необхідності порядкові номери рядків вказують безпосередньо перед їх назвами. Перед числовими значеннями величин порядкові номери в таблицях не проставляються.

Якщо текст, що повторюється в графі таблиці, складається з одного слова, то його можна заміняти лапками; якщо з двох або більше слів, то при першому повторенні його замінюють словами «Те ж», а далі лапками. Ставити лапки замість повторюваних цифр, марок, знаків, математичних і хімічних символів не дозволяється. Якщо цифри чи інші дані в рядку таблиці не наводяться, то в ньому ставлять прочерк.

Якщо таблиця виходить за рамки сторінки, її поділяють на частини, розміщуючи одну під іншою або поруч, або переносять частину таблиці на наступну сторінку. У кожній частині таблиці повторюють її головку та боковик. У разі поділу таблиці на частини дозволено її головку чи боковик заміняти відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами в першій частині таблиці. Слово «Таблиця» пишуть один раз над першою частиною таблиці. Над іншими частинами таблиці з абзацного відступу друкують, наприклад, «Продовження таблиці 2.2» або «Кінець таблиці 2.2», не повторюючи її назви. Якщо в кінці сторінки таблиця переривається й продовжується на наступній сторінці, то в першій частині таблиці нижню горизонтальну лінію, що обмежує таблицю, не проводять.

Таблиці з невеликою кількістю колонок дозволено поділяти на частини й розміщувати частини поруч на одній сторінці, повторюючи головку таблиці. У цьому випадку рекомендується розділяти частини таблиці подвійною лінією.

## ***3.7 Переліки***

Переліки (за потреби) подають у розділах, підрозділах, пунктах і/або підпунктах. Перед переліком ставлять двокрапку (крім пояснювальних переліків на рисунках). Якщо перелік має один рівень підпорядкованості, і на пункти переліку немає посилань, то перед кожним пунктом ставлять тире. Якщо в тексті є посилання на пункти переліку або перелік має більше одного рівня, то пункти верхнього рівня позначають малими літерами української абетки, далі – арабськими цифрами, далі – знаками тире. Після цифри або літери, якою починається позиція переліку, ставлять круглу дужку.

**Приклад**

Перелік з трьох рівнів:

а) перший пункт першого (верхнього) рівня;

б) другий пункт першого (верхнього) рівня;

1) перший пункт другого рівня;

– перший пункт третього рівня;

– другий пункт третього рівня;

2) другий пункт другого рівня;

в) третій пункт першого (верхнього) рівня.

У разі складної ієрархії переліків дозволено користуватися можливостями текстових редакторів автоматичного створення нумерації переліків. Текст кожної позиції переліку треба починати з малої літери з абзацного відступу відносно попереднього рівня підпорядкованості.

## ***3.8 Примітки***

Примітки подають, якщо є потреба пояснень до тексту, таблиць, рисунків. Примітки записують безпосередньо після абзаца, під рисунком (перед його назвою) або під таблицею, яких стосується примітка. Слово «Примітка» друкують кеглем 12 через один міжрядковий інтервал з абзацного відступу з великої літери з крапкою в кінці. У тому самому рядку через проміжок з великої літери друкують текст примітки тим самим шрифтом. Одиночну примітку не нумерують.

**Приклад**

Примітка. Текст одиночної примітки.

Якщо приміток дві або більше, то їх нумерують арабськими цифрами.

**Приклад**

Примітка 1. Текст першої примітки.

Примітка 2. Текст другої примітки.

## ***3.9 Виноски***

Пояснення до окремих даних, наведених у тексті або таблиці, можна оформлювати як виноски. Виноски позначають над рядком арабськими цифрами з круглою дужкою, наприклад 1), і нумерують у межах кожної сторінки. Дозволено виноску позначати зірочкою (\*). На одній сторінці тексту може бути не більше ніж чотири виноски.

Позначку виноски ставлять після слова, числа, символу або речення, до якого дають пояснення. Цю ж позначку ставлять перед пояснювальним текстом.

Виноски пишуть з абзацним відступом: позначену в тексті – внизу сторінки з позначкою, в таблиці – під основною частиною таблиці. Виноску відокремлюють від основного тексту чи таблиці тонкою горизонтальною лінією завдовжки 30–40 мм з лівого берега. Текст виноски має кегль 12 і один міжрядковий інтервал.

## ***3.10 Формули***

Формули (включно з рівняннями) друкують посередині сторінки симетрично тексту окремим рядком безпосередньо після тексту, в якому їх згадано. Верхній та нижній записи формули мають бути на відстані не менше ніж один рядок від попереднього та подальшого тексту.

Формули набираються в редакторі *MathType* або *Equation Editor* з такими шрифтами та стилями.

Текст (Text) – Times New Roman.

Функції (Function) – *Times New Roman*.

Змінні (Variable) – Times New Roman, курсив.

Грецькі символи (Greek) – Symbol, курсив.

Символи (Symbol) – Symbol.

Вектори та матриці (Vector-Matrix) – Times New Roman, жирний.

Числа (Number) – *Times New Roman.*

Розміри шрифтів мають бути такими.

Основний розмір (Full) – 14 pt.

Індекси першого рівня (верхні та нижні) (Subscript/Superscript) – 58 % від основного розміру.

Індекси другого рівня (верхні та нижні) (Sub-Subscript/Superscript) – 42 % від основного розміру.

Розмір символів (Symbol) – 150 % від основного розміру.

Символи-індекси (Sub-symbol) – 100 % від основного розміру.

Нумерують лише ті формули, на які є посилання в тексті. Формули в тексті (не в додатку), нумерують арабськими цифрами в круглих дужках – або наскрізно, або в межах кожного розділу – (2) або (1.2). У додатку номер формули складається з великої літери, що позначає додаток, і порядкового номера формули або рівняння в цьому додатку, відокремлених крапкою: (А.3).

Нумеровані формули розміщуються в окремих рядках (рамки таблиці встановлюються невидимими). Номери формул друкують на їх рівні праворуч у крайньому положенні. У багаторядкових формулах номер записують на рівні останнього рядка. Якщо в тексті або в додатку є лише одна формула, її нумерують (1) або (А.1).

Формула є частиною речення, тому до неї застосовують звичайні правила граматики. Зокрема, якщо формула знаходиться в кінці речення, то після неї ставлять крапку.

За необхідності позначення у формулі пояснюють з нового рядка без абзацного відступу, починаючи зі слова «де» без двокрапки. Пояснення вирівнюють по вертикалі.

**Приклад**

Відомо, що ,

|  |  |
| --- | --- |
|  | (3.1) |

де М1, М2 – математичне очікування;

σ1, σ2 – середні квадратичні відхили [23].

Верхні та нижні індекси у формулах, а також показники степеня, в усьому тексті мають бути однакового розміру й меншими ніж символи, яких вони стосуються.

Переносити формули на наступний рядок можна лише на знаках операцій, які пишуть у кінці попереднього рядка та на початку наступного. У перенесенні формули на знакові операції множення застосовують знак «х». Перенесення на знаку ділення «:» слід уникати.

Кілька формул підряд, не відокремлених текстом, пишуть одну під одною й розділяють комами.

**Приклад**

Об’єми утворених фігур виражаються такими формулами:

|  |  |
| --- | --- |
| *V*1 = *f1(x, y, z),* | (4.25) |
| *V2* = *f2(x, y, z).* | (4.26) |

Числові значення величин з допусками наводять у такому вигляді: (65º±º3) %, 80 мм ± 2 мм або (80 ± 2) мм.

Діапазон чисел фізичних величин наводять, використовуючи прикметники «від» і «до», наприклад, «… від 1 мм до 5 мм» (а не «… від 1 до 5 мм»).

Два чи три виміри вказують, наприклад «80 мм х 25 мм х 50 мм» (а не «80ºхº25 х 50 мм»).

## ***3.11 Посилання***

У тексті можна робити посилання на структурні елементи самого тексту (розділи, підрозділи, пункти, підпункти, позиції переліків, рисунки, формули, рівняння, таблиці, додатки) або на інші джерела.

У посиланні на елементи самого тексту вказують їх номери в таких виразах: «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. Дозволено використовувати загальноприйняті та застандартовані скорочення згідно з ДСТУ 3582, наприклад, «згідно з рис. 10», «див. табл. 3.3» тощо. Посилання на рисунки й таблиці, які наведено вище, дають зі скороченим словом «дивись» у дужках: «(див. рис. 1.3)», (див. табл. 3.1).

Посилаючись на позицію переліку, записують номер структурного елемента тексту та номер позиції переліку з круглою дужкою, відокремлені комою. Якщо переліки мають кілька рівнів, то їх вказують, наприклад так: «відповідно до 2.3.4.1, б), 2)».

Посилання на джерело інформації, наведене в переліку джерел посилання, рекомендовано подавати тим номером у квадратних дужках, за яким це джерело зазначено в переліку джерел посилання, наприклад, «у статті [1]», «у роботах [3] – [5]». Нумерація посилань на джерела інформації починається з 1.

Джерело інформації можна подати у виносці, навівши номер і опис джерела як у переліку джерел посилання. Наприклад, текст містить номер джерела у дужках [5] й позначення виноски: «Докладніше питання алгоритмічної теорії великих чисел висвітлено в [5].1)». Тоді у виносці після номера зміст повторює запис джерела з номером 5 у переліку джерел.

1) [5] Анісімов А. В. Алгоритмічна теорія великих чисел / А. В. Анісімов. – Київ: Видавничий дім «Академперіодика», 2001. – 153 с.

## **3.12 Додатки**

Додатки розміщують у порядку посилання на них у тексті. Кожен додаток повинен мати заголовок, який друкують вгорі малими літерами з першої великої симетрично до тексту сторінки. Над заголовком посередині рядка друкують слово **ДОДАТОК** і відповідну велику літеру української абетки, крім літер Ґ, Є, З, І, Ї, Й, О, Ч, Ь, яка позначає додаток, наприклад **ДОДАТОК А**, **ДОДАТОК Б**. Дозволено позначати додатки літерами латинської абетки, крім літер І та О. У разі повного використання літер української і/або латинської абеток дозволено позначати додатки арабськими цифрами. Один додаток позначають як **ДОДАТОК А**.

Кожен додаток починають з наступної сторінки. Нумерація аркушів документа і його додатків наскрізна.

Якщо додаток є документом, що має самостійне значення (наприклад, патентні дослідження, технічні умови, технологічний регламент, атестовану методику проведення досліджень, стандарт тощо) й оформлений згідно з вимогами до цього документа, то в додатку вміщують його копію без будь-яких змін. Перед копією розміщують окремий аркуш, на якому друкують великими літерами слово **ДОДАТОК** і відповідну велику літеру, що позначає додаток, а під ними, симетрично відносно сторінки, – назву документа малими літерами, починаючи з першої великої. Аркуш із цими даними нумерують. На копії документа-додатка у правому верхньому куті продовжують нумерацію сторінок роботи, а знизу зберігають нумерацію сторінок документа.

За потреби текст додатків поділяють на розділи, підрозділи, пункти й підпункти, які нумерують в межах додатка відповідно до вимог у п. 4.4. Перед кожним номером ставлять позначення додатка (літеру) і крапку, наприклад, **А.2** – другий розділ додатка А, **Б.2.1** – підрозділ 2.1 додатка Б, **В.3.1.2** – пункт 3.1.2 додатка В, **Г.1.3.3.4** – підпункт 1.3.3.4 додатка Г.

Рисунки, таблиці та формули в тексті додатків нумерують у межах кожного додатка, починаючи з літери, що позначає додаток, наприклад, рисунок А.2 – другий рисунок додатка А, таблиця Б.2 – друга таблиця додатка Б, формула (В.1) – перша формула додатка В. Якщо в додатку один рисунок, одна таблиця чи одна формула, то їх нумерують, наприклад, рисунок А.1, таблиця Б.1, формула (В.1).

Посилання в тексті додатка на рисунки, таблиці, формули, рівняння подають згідно з п. 4.11. Переліки, примітки та виноски в тексті додатка оформляють і нумерують згідно з п. п. 4.7, 4.8, 4.9.

Джерела, цитовані лише в додатку, розглядаються незалежно від цитованих в основній частині роботи і наводяться в кінці цього додатка в переліку джерел посилання. Форма цитування, правила складання переліку джерел посилання та виносок у додатках аналогічні прийнятим в основній частині звіту. Перед номером цитати та відповідним номером у переліку джерел посилання й виносках ставлять позначення додатка.

## **3.13 Оформлення коду**

### **3.13.1. Форматування коду**

Код повинен бути відформатований відповідно до наступних правил:

а) параметр "Tabs" повинен бути встановлений в положення "Keep spaces";

б) розмір відступу повинен бути 4 пробілу;

в) програмні конструкції вибору та повторення повинні бути відформатовані, як в наступних прикладах:

if (file.Exists(fileName))

{

file.Open(fileName);

}

for (int i = 0; i < MAX\_ELEMENTS; ++i)

{

array[i] = i \* MULTIPLY\_FACTOR;

}

switch (workMode)

{

case WorkMode.Add:

DoAdd();

break;

case WorkMode.Update:

DoUpdate();

break;

}

г) Порожні рядки використовуються для поліпшення сприйняття коду. Зазвичай вони використовуються для поділу незалежних секцій коду. Обов'язково розміщуйте порожній рядок:

1) після останнього оголошення "using" перед оголошенням простору імен;

2) між методами;

3) між оголошенням локальних змінних методу та першої інструкцією методу;

4) перед багаторядковим та однорядковим коментарем за винятком коментаря після фігурної дужки, що відкриває блок;

5) перед логічно ізольованій частиною коду методу.

д) пробіли також слід використовувати для поліпшення сприйняття коду. Необхідно вставляти пробіли:

– перед круглими дужками, наступними за ключовим словом:

if (condition)

– після коми в списку аргументів:

int result = Calculate(argumentOne, **argumentTwo**);

– між бінарним оператором та його операндами:

int result = **argumentOne + argumentTwo**;

– між частинами інструкції "for":

for (int i = 0; i < MAX\_ELEMENTS; ++i)

е) Аргументи унарних операторів ніколи не відокремлюються пробілами:

i++;

--counter;

ж) Всі довгі рядки повинні бути згорнуті. Довжина рядка не повинна перевищувати 80 символів. При згортанні довгих рядків намагайтеся дотримуватися наступних правил:

1) переносите рядок після коми;

2) переносите рядок перед оператором;

3) віддавайте перевагу «логічному» розбиттю рядка, а не «фізичному»;

4) Робіть додатковий відступ перед «згорнутої» частиною рядка:

int i;

do

{

// Here comes a long line.

bool isSucceeded = a + b + c + d + e – h \* MAX\_FACTOR / DEFAULT\_SCALE

- **correctionFactor;**

}

з) Довгий список параметрів методу також повинен бути згорнутий, так само як та список аргументів у інструкції виклику методу:

// The method declaration.

public float DoSomethingFromManyArguments(

**int argumentOne,**

**int argumentTwo,**

**decimal argumentThree,**

**string argumentFour)**

{

// Method body goes here.

}

// The calling code.

float result = DoSomethingFromManyArguments(

**1,**

**1,**

**0.5m,**

**“Some text”**);

і) Не оголошуйте декілька змінних в одному рядку, використовуйте окремий рядок для кожної змінної:

int customerID;

string customerName;

float salary;

### **3.13.2 Угоди про найменування**

Не використовуйте угорську або іншу префіксних нотацію за винятком випадків, обумовлених далі.

Імена локальних змінних та аргументів методів повинні записуватися в нотації «верблюд»:

string customerName;

private int DoSomething(int **firstArgument,** float **secondArgument**);

Імена закритих полів класу повинні слідувати нотації «верблюд»:

private string **customerName**;

Імена відкритих полів класу повинні слідувати нотації «Паскаль»:

public int **CustomerID;**

Імена властивостей та методів класу повинні слідувати нотації «Паскаль»:

public bool **ValidateAmount**();

public bool **FileExists**

{

get;

}

Імена класів, структур та нумераторів повинні слідувати нотації «Паскаль»:

public class **CustomerAccount**

{

};

private enum **WorkMode**

{

**Add,**

**Update**;

}

Імена методів повинні слідувати шаблоном «дієслово» + «іменник» - наприклад, “UpdateAccount”;

Використовуйте однину, а не множину в іменах перечислення. Іншими словами,“WorkMode” - це правильне ім'я, а "WorkModes" - ні;

Використовуйте значущі імена навіть для закритих методів, властивостей, типів та інше. Уникайте використовувати короткі імена, такі як “a”, “b”, “n” за винятком загальноприйнятих імен для змінних циклу “i” та “j”.

### **3.13.3 Загальні принципи розробки**

Культура кодування:

а) Дублювання коду в програмі суворо забороняється.

б) Обов'язкова заміна числових та строкових літералів символічними константами. Виняток можна зробити для самоочевидних констант, таких як 0 та 1. Давайте символічним константам значущі імена;

в) Рекомендується об'єднувати пов'язані між собою константи цілих типів в нумератори, а не цілих типів - в абстрактні класи з відкритими статичними константними членами;

г) Уникайте довгих та складних методів, розбивайте їх на декілька коротких. При модифікації коду максимально використовуйте рефакторинг;

д) Завжди звертайте увагу на попередження компілятора. Подбайте, щоб їх не було.

Угоди про коментарі.

1. Обов'язкові коментарі до класів та до інших призначених для користувача типів та до всіх відкритих членів класів.
2. Коментарі всередині коду методів не обов'язкові. Код повинен легко читатися та без коментарів.
3. Коментар не повинен перефразовувати те, що написано в коді. Намагайтеся давати змістовні пояснення. Наприклад, коментар “Increment i by one” до коду “i++;” це погана практика, а коментар “Update the number of customer accounts processed” – добра.
4. Текст коментаря відокремлюйте від слешів одним пропуском. Перша буква речення повинна бути прописною, в кінці речення повинна стояти крапка:

// This is a comment. This is yet another comment sentence.

1. Коментарі повинні підкорятися загальним правилам згортання рядків.
2. Не використовуйте коментарі в стилі C: /\* … \*/

# **4 Застереження щодо плагіату в курсовому проекті.**

Зазвичай при написанні КР студентами використовуються різні наукові джерела, але потрібно пам‘ятати, що пряме запозичення тексту без посилання на джерело або завуальоване, коли текст з джерела зазнає несуттєвих змін шляхом заміни окремих слів та виразів їх синонімічними аналогами і видається за власний, розцінюється як плагіат, який за своєю суттю є викраденням результатів чужої інтелектуальної праці [9].

Для уникнення звинувачень в плагіаті при прямому цитуванні або опосередкованому обов‘язково вказувати посилання на джерело (див. 3.11). При цитуванні можна використовувати фрази "Як зазначає проф. ІвановºП.ºМ., …", "На думку проф. Іванова П. М., …", "Проф. Іванов П. М. вважає, що …", "Необхідно погодитись із … у тому, що …". При прямому цитуванні текст потрібно наводити дослівно, включно з помилками і знаками пунктуації. Пряме цитування бажано звести до мінімуму і вдаватися до нього лише у випадках, коли воно справді необхідне.

При цитуванні потрібно дотримуватися таких правил:

1. Цитата повинна бути дослівною.

2. Цитата не повинна бути надто довгою.

3. Цитата повинна подаватися у лапках і супроводжуватися посиланням на джерело з переліку використаних джерел.

4. Вилучення певних фрагментів, яке дозволяє уникнути довгого цитування, позначається трикрапкою ("....").

5. Пояснення, які розривають текст цитати, подаються у квадратних дужках - [ ];

6. Цитати, які є незавершеним реченням, граматично узгоджуються із авторським текстом.

Замість прямого цитування можна використовувати перефразовування (переказування). При цьому потрібно прагнути передати думку автора як найближче до оригіналу з використанням власної лексики, а для уникнення плагіату дотримуватись таких підходів: виділити основні ідеї, змінити структуру речень, використовувати синоніми або слова з подібним значенням, змінювати форми слів. При цьому наприкінці речень обов’язково мають бути посилання [10].

# **5 Захист та оцінювання курсової роботи**

## ***5.1 Захист курсової роботи***

Захист курсової роботи відбувається у відповідності до графіку захисту курсових робіт у наступному порядку:

* студент протягом 5 – 10 хвилин викладає зміст своєї роботи (можна використовувати технічні засоби);
* студент відповідає на питання, запропоновані членами комісії;
* зачитується відгук керівника;
* студент відповідає на зауваження, що є у відгуку;
* керівник та члени комісії можуть виступити за матеріалами захисту;
* результати захисту повідомляються керівником після обговорення.

## ***5.2 Оцінювання курсової роботи***

За курсову роботу виставляється диференційований залік у відомість та залікову книжку студента. При оцінюванні враховується:

* актуальність теми;
* теоретичний рівень роботи;
* практична цінність роботи;
* якість оформлення;
* виступ на захисті;
* рекомендація керівника.

Курсова робота оцінюється за 100-бальною шкалою і за національною шкалою оцінок, наведеною у табл.2.

Таблиця 2 - Шкала оцінювання

| 100-бальна система | Національна шкала | |
| --- | --- | --- |
| 90-100 | відмінно | 5 |
| 75-89 | добре | 4 |
| 60-74 | задовільно | 3 |
| 1-59 | не задовільно | 2 |

При наявності невчасного виконання етапів курсової роботи відповідно до календарного плану-графіку (Додаток1) можуть зніматись бали — по одному за кожний тиждень запізнення (із 100 балів).

# **Перелік використаних джерел.**

1. Методичні вказівки з підготовки та оформлення кваліфікаційних та курсових робіт для студентів факультету комп’ютерних наук та кібернетики / Л. Л. Омельчук, А. Б. Ставровський – К.: Київський національний університет імені Тараса Шевченка, 2017 – 47 с.
2. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання [Текст] : ДСТУ 3008:2015. – К. : ДП «УкрНДНЦ», 2016. – V, 26 с. – (Національний стандарт України).
3. **ДСТУ ГОСТ 7.1:2006. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання** / Нац. стандарт України. – Вид. офіц. – [Чинний від 2007-07-01]. – Київ : Держспоживстандарт України, 2007. – 47 с. – (Система стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України).
4. ДСТУ 4302:2004. Інформаційні технології. Настанови щодо документування комп’ютерних програм. - К.: Держстандарт України, 2004.
5. ДСТУ ISO/IEC/IEEE 15289:2019 (ISO/IEC/IEEE 15289:2017, IDT) Інженерія систем і програмних засобів. Уміст інформаційних об’єктів життєвого циклу. - К.: Держстандарт України, 2019.
6. **ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання** / Нац. стандарт України. – Вид. офіц. – [Уведено вперше ; чинний від 2016-07-01]. – Київ : ДП «УкрНДНЦ», 2016. – 17 с.
7. Петренко І. І. Питання виявлення плагіату літературного твору в умовах розвитку / І. І. Петренко // Теоретичні і практичні аспекти економіки та інтелектуальної власності. — Науково - дослідний інститут інтелектуальної власності, м. Київ — 2009. — С. 57 - 60.
8. Шейко В.М., Кушнаренко Н.М. Організація та методика науково- дослідницької діяльності. Підручник. - К., 2003. - 295 с.
9. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html> (дата звернення: 13.11.2023)
10. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). – Addison-Wesley Professional, 2003. – 208 p. ISBN-10: 9780321193681, ISBN-13: 978-0321193681.

# **Додаток 1. Календарний план графік**

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**

**виконання курсової роботи з дисципліни**

**«Вступ до об‘єктно-орієнтованого програмування»**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Термін закінчення виконання етапів роботи | Примітка |
| 1. | Затвердження теми курсової роботи |  |  |
| 2. | Ознайомлення з ЄСПД, ЄСКД, ДСТУ 3008-95 |  | 2 дні |
| 3. | Вивчення та аналіз предметної області об'єкту курсової роботи |  | 5 днів |
| 4. | Розробка архітектури та загальної структури системи |  | 5 днів |
| 5. | Розробка Розділу І |  |  |
| 6. | Розробка розділу ІІ |  |  |
| 9. | Оформлення пояснювальної записки |  | 4 дн.  Результат -друкований примірник |
| 10. | Підготовка до захисту курсової роботи |  | 1 дн. |
| 11. | Захист курсової роботи |  | 3дн. |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Прізвище, ім'я та по-батькові керівника |
| Студент |  | Прізвище, ім'я та по-батькові студента |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

дата затвердження

# **Додаток 2. Зразок титульного аркушу до курсової роботи**

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ

«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУТ ІМЕНІ ТАРАСА ШЕВЧЕНКА»

Циклова комісія програмування та інформаційних технологій

**КУРСОВА РОБОТА**

**з дисципліни «Вступ до об‘єктно-орієнтованого програмування»**

на тему \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| ЗАХИЩЕНО З ОЦІНКОЮ  Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_  Кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Керівник  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ПІБ керівника  (підпис) |  | ВИКОНАВ |
| студент 3-го курсу групи ІПЗ-31 спеціальність ***121*** «Інженерія програмного забезпечення» |
| Прізвище Ім'я По-батькові  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця) |

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

(підпис)

м. Київ — 2024

# **Додаток 3. Зразок оформлення змісту**

**ЗМІСТ**

|  |  |
| --- | --- |
|  | стор. |
| Реферат …………………………………………………………………... | 3 |
| Перелік скорочень ……………………………………………………… | 4 |
| Вступ ……………………………………………………………………. | 5 |
| Розділ 1 НАЗВА РОЗДІЛУ ……………………………………………. | 6 |
| 1.1 Назва пункту ……………………………………………………… | 6 |
| 1.2 Назва пункту ……………………………………………………… | 10 |
| 1.2.1 Назва підпункту ………………………………………………. | 12 |
| 1.2.2 Назва підпункту ………………………………………………. | 15 |
| Розділ 2 НАЗВА РОЗДІЛУ …………………………………………….. | 20 |
| 2.1 Назва пункту ………………………………………………………. | 22 |
| 2.2 Назва пункту ………………………………………………………. | 25 |
| ВИСНОВКИ …………………………………………………………….. | 30 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ………………………………. | 32 |
| Додаток 1 Назва додатку ……………………………………………….. | 33 |
| Додаток 2 Назва додатку ……………………………………………….. | 34 |

# **Додаток 4. Зразок оформлення реферату**

**РЕФЕРАТ**

Курсова робота складається із вступу, двох розділів, висновків, переліку використаних джерел (23 найменування) та одного додатку. Робота містить два рисунки та 3 таблиці. Загальний обсяг роботи становить 35 сторінок, основний текст викладено на 28 сторінках.

Ключові слова: КЛЮЧОВЕ СЛОВО 1, КЛЮЧОВЕ СЛОВО 2, …, КЛЮЧОВЕ СЛОВО N.

В роботі досліджено… Отримано нові … Впроваджено …

# **Додаток 5. Зразок оформлення прийнятих скорочень**

**Перелік скорочень, умовних позначень, символів, одиниць і термінів**

|  |  |  |
| --- | --- | --- |
| ОС | — | операційна система |
| ООП | — | об‘єктно – орієнтоване програмування |
| ПЗ | — | програмне забезпечення |
|  |  |  |
|  |  |  |

# **Додаток 6. Зразок оформлення використаних джерел**

**Перелік використаної літератури**

1. Бандоріна Л.М., Климкович Т.О., Удачина К.О. Основи алгоритмізації та програмування : навч. посібник. УДУНТ, 2022. 158 с. URL: http://eadnurt.diit.edu.ua/bitstream/123456789/15729/1/Bandorina.pdf (дата звернення: 13.02.2019).
2. Боровльова С. Ю. Базовий С++ : навчальний посібник /С. Ю. Боровльова, А. В. Швед. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017.
3. Вступ до програмування мовою С++. Організація обчислень : навч. посіб. / Ю. А. Бєлов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. ISBN (укр.). URL: http://csc.knu.ua/en/library/books/belov-24.pdf (дата звернення: 13.02.2019)
4. Ковалюк Т. В. Алгоритмізація та програмування: Підручник. — Львів: «Магнолія 2006», 2013. — 400 с., ил
5. Конституція України (Відомості Верховної Ради України (ВВР), 1996, № 30, ст. 141) {Із змінами, внесеними згідно із Законами № 2222-IV від 08.12.2004, ВВР, 2005, № 2, ст. 44, № 2952-VI від 01.02.2011, ВВР, 2011, № 10, ст. 68, № 586-VII від 19.09.2013, ВВР, 12 2014, № 11, ст. 142, № 742-VII від 21.02.2014, ВВР, 2014, № 11, ст. 143, № 1401-VIII від 02.06.2016} // Вища рада правосуддя. Офіційний сайт. Нормативні акти. URL: http://www.vru.gov.ua/legislative\_acts/1 (дата звернення: 13.02.2019).
6. Основи об’єктно-орієнтованого програмування : навч. посібник / Гришанович Т. О., Глинчук Л. Я.; ВНУ імені Лесі Українки. Луцьк : ВНУ імені Лесі Українки, 2022. – 120 с. URL: https://evnuir.vnu.edu.ua/bitstream/123456789/20320/1/oop.pdf
7. **Статистичний** щорічник за 2010 рік [Електронний ресурс] / Держ. служба статистики України. – Київ : Август Трейд, 2011. – Режим доступу : [http://library.oseu.edu.ua/docs/StatSchorichnyk Ukrainy 2010.pdf](http://library.oseu.edu.ua/docs/StatSchorichnyk%20Ukrainy%202010.pdf). – Назва з екрану (дата звернення: 21.10.2020
8. Kanungo, T. An efficient k-means clustering algorithm: Analysis and implementation / T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. – Vol. 24, no. 7. - PP. 881-892.
9. Мар’їна О. Контент-стратегія бібліотек у цифровому середовищі Бібліотечний вісник. 2016. № 4. С. 8–12. URL: http://nbuv.gov.ua/UJRN/bv\_2016\_4\_4 (дата звернення: 26.09.2017)..

# **Додаток 7. Критерії оцінювання курсової роботи**

При оцінюванні курсової роботи слід зважати на те, що здобувач  
повинен уміти:

- працювати з інформаційними джерелами (законодавчою і нормативною базою, науковою спеціальною літературою, у т.ч. на іноземній мові, матеріалами глобальних інформаційних мереж, даними статистичної звітності тощо);

- логічне та аргументовано викладати матеріал, обґрунтовувати проектні рішення за видами забезпечення;

- робити висновки.

Оцінювання якості підготовки курсової роботи здійснюється за 100-бальною шкалою за модульно-рейтинговою системою. Максимальний рейтинг складається із середнього арифметичного оцінювання в балах за всіма критеріями комісією, які надані в таблиці нижче, та виставляється під час захисту курсової роботи і переводиться в оцінку за схемою нарахування рейтингу:

90-100 балів – відмінно;

70-89 балів – добре;

50-69 балів – задовільно;

менше 50 балів – незадовільно.

Таблиця - Критерії оцінювання курсової роботи комісією

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Пояснювальна  записка | Ілюстративна  частина (презентація) | Захист  роботи | Відповіді на  додаткові  питання | Сума |
| До 60 | До 10 | До 20 | До 10 | 100 |

Оцінка може знижена за недотримання графіка розробки курсового проекту (5 балів), зокрема за несвоєчасне завершення виконання проекту і подання його на повторний захист (оцінка повторного захисту не може перевищувати 60 балів). Також бали знімаються тоді, коли у пояснювальній записці недостатньо розкрито зміст певних розділів курсового проект.

Додаткові бали можуть бути надані за участь студента у захисті робіт інших студентів (до 10 балів). Студенти отримують бали за цим критерієм тоді, коли під час захисту курсових проектів іншими студентами вони задають їм додаткові запитання, а також вносять свої пропозиції щодо вдосконалення розроблених програм. За кожне правильно та обґрунтовано задане запитання студент може отримати від 1 до 2 балів.

Оформлення пояснювальної записки.

Максимальна оцінка по цьому критерію виставляється тоді, коли пояснювальна записка відповідає усім вимогам, встановленим у методичних вказівках до курсової роботи, а також коли записка оформлена акуратно і відповідно до стандартів. Особлива увага при цьому приділяється оформленню малюнків і таблиць.

Підготовка ілюстративної частини.

До захисту студент готує доповідь, яку супроводжує ілюстративний матеріал у вигляді презентації. Поданий до захисту ілюстративний матеріал повинен достатньою мірою відображати хід виконання курсової роботи і результати проектування. Презентація оцінюється за повнотою висвітлення відповідних положень доповіді.

Доповідь студента під час захисту проекту

Даний критерій здатність студента вільно виступати перед аудиторією, доповідаючи про процес розробки та особливості роботи розробленого ним проекту, користуючись набутими у коледжі знаннями із різних дисциплін. Комісією в першу чергу оцінюється ґрунтовність та лаконічність відповіді, послідовність викладення матеріалу, вміння оперувати основними термінами та поняттями. Максимальний бал студент отримує, якщо під час доповіді студент вільно володіє термінологією, орієнтується у всіх деталях роботи над проектом, а також у предметній області, для якої цей проект розроблявся.

Відповіді студента на додаткові запитання комісії.

Студент отримує максимальний бал по цьому критерію, якщо він надав вичерпні та точні відповіді на додаткові запитання комісії, які стосуються розробленого ним продукту, основних термінів програмної інженерії, які використовувались для розробки проекту, а також основних понять та термінів із дисциплін та галузей науки, до яких належить предметна область розробленого студентом проекту.

# **Додаток 8. Порядок проведення захисту курсової роботи**

Курсова робота має бути здана керівнику на перевірку в термін, визначеному в індивідуальному завданні. Курсова робота допускається до захисту за умови закінченого оформлення (наявність пояснювальної записки і її електронної версії на CD або DVD-диску разом з первісними текстами програми, ЕХЕ-файлом та вхідними і тестовими даними) і допуску наукового керівника або наявності рецензії (письмового відгуку).

Порядок захисту курсових робіт передбачає:

1) доповідь студента про хід виконання курсової роботи і головні отримані результати;

2) відповіді на запитання членів комісії.

Захист курсових робіт відбувається до початку екзаменаційної сесії. Порядок захисту, визначається керівником і доводиться до студентів не пізніше, ніж за тиждень до захисту. Захист курсових робіт проводиться у встановлений час у вигляді публічного виступу студента: захисту перед комісією за участю керівника роботи. За відсутності керівника курсової роботи захист може бути проведена за умови подання ним письмового відгуку на курсову роботу.

Для захисту курсової роботи студенту надається 7-10 хв. Для викладу основних положень роботи. В ході захисту курсової роботи необхідно продемонструвати весь процес проектування. Студенти, які в термін, відведений на виконання курсової роботи, не оформили пояснювальну записку і не здали її керівнику на перевірку, до захисту курсової роботи не допускаються і отримують оцінку "незадовільно". До повторного захисту курсової роботи вони допускаються після оформлення пояснювальної записки і її перевірки. У студентів, що не захистили свої роботи або не з'явилися на захист, повторний захист може проводитися в період екзаменаційної (залікової) сесії.

Критеріями оцінки курсової роботи є:

− актуальність і ступінь розробленості теми;

− творчий підхід і самостійність в аналізі, узагальнення та висновки;

− повнота охоплення першоджерел та дослідницької літератури;

− якість проектних рішень та документації;

− науковий стиль викладу;

− дотримання всіх вимог до оформлення курсової роботи і термінів її виконання;

− аргументованість і точність відповідей на питання.

Захист курсової роботи відбувається прилюдно на відритому засіданні комісії у складі провідних викладачів циклової комісії з програмування та інформаційних технологій. При оцінюванні курсової роботи враховуються: якість виконаного проекту (роботи), ступень самостійності роботи автора і проявлена ним ініціатива; оформлення курсової роботи, якість проведених проектних рішень, графічних робіт, оригінальність, зв’язність викладання і вміння висловлювати думки, володіння науково-технічною термінологією спеціальності; теоретична і практична підготовка з дисциплін, передбачених навчальним планом.

Доповідь студент повинен підготувати заздалегідь у формі виступу, в якому доцільно висвітлити такі питання:

- тема курсового проекту;

- обґрунтування актуальності теми дослідження/проекту;

- мета, завдання, об’єкт, предмет дослідження/проекту;

- довести основні результати та методи, за допомогою яких вони були досягнуто;

- стислі пояснення роботи основних частин структурних схем,

- елементи новизни у теоретичних положеннях та в практичних рекомендаціях;

- особливості розробленої системи, висновки;

Під час доповіді студент повинен обов’язково сказати про поданий графічний матеріал.

Рішення про оцінювання курсової роботи приймається комісією. Здобувач, який не захистив курсову роботу допускається до повторного захисту курсової роботи після усунення недоліків у терміни визначені цикловою комісією.

Студент повинен добре знати зміст курсової роботи і бути готовим відповідати на будь-яке питання за змістом доповіді та текстом пояснювальної записки, а також на питання, що належать до загальних принципів роботи основних частин проекту або системи.

# **Додаток 9. Рекомендації щодо написання доповіді та створення презентації до захисту курсової роботи, підготовки матеріалів у курсовій роботі**

Рекомендації щодо написання доповіді

Для представлення наукових та практичних результатів потрібно підготувати доповідь. Зміст доповіді залежить від характеру роботи. Так, для представленні курсової роботи на захисті доповідь має таку структуру:

• Шановні члени комісії та присутні, Вашій увазі пропонується курсова  
робота на тему: «...». Виконавець ..., керівник ...;

• короткий виклад мети, об’єкту дослідження/проектування, задач роботи, визначення методів та інструментів дослідження/проектування та розроблення;

• послідовний виклад результатів курсової роботи: обрані алгоритми та методи реалізації задач, UML-діаграми реалізації проекту, структурні схеми програмного продукту та його архітектурних рішень, граф-схеми алгоритмів з коментарями їх особливостей (всі положення ілюструються схемами, алгоритмами тощо з пояснювальної записки),

• коротка характеристика створеного проекту програмного продукту (результату курсової роботи), функціональні можливості, інтерфейс користувача,

• практичне застосування результатів роботи, а також перспективи подальшого розвитку програмного продукту (якщо такі передбачаються).

• Доповідь закінчена. Дякую за увагу!

Доповідь розраховують на 4-5 хвилин. Доповідач має підготуватися до ймовірних запитань по курсовій роботі. Деякі з них він викликає сам, коротко говорячи про результати (але інакше в межах виділеного часу неможливо). Інші виникають по причині неякісного опрацювання теоретичного матеріалу чи неповного виконання завдання. Студент має підготуватися до відповіді на такі питання. Якщо не знаєте відповіді на поставлене запитання і цього не вимагалося індивідуальним завданням, можна сказати (там де це коректно), що така задача не ставилася.

Рекомендації щодо створення презентації

Для викладення доповіді доповідач створює презентацію, яка містить ілюстративний матеріал до доповіді. Вона складається з послідовності слайдів, які можуть викладатися у наступній послідовності:

*1 слайд*. Тема курсової роботи, виконавець та керівник.

*2 слайд*. Мета та задачі, які були поставлені на курсову роботу.

*3 слайд*. Коротко про предметну область.

*4 ÷ n сл*айд. Демонстрація отриманих результатів (за розділами доповіді), це можуть бути текстові блоки, таблиці, блок-схеми алгоритмів, UML-діаграми, структурні схеми тощо. На одному слайді краще не розміщувати більше 8-10 рядків тексту та 1-2 схем (блок-схем). Кожен слайд повинен мати заголовок.

*Останній слайд*. Висновки за отриманими результатами (коротко).

*Додатковий слайд* (не обов’язковий). Перспективи і проблеми, які  
опрацьовує виконавець для подальшого розвитку теми роботи.

Наприкінці доцільно зробити слайд "Дякую за увагу!"

Вимоги до оформлення презентації:

1) Обсяг презентації має бути не менше 15 та не більш 30 слайдів.

2) Презентація включає наочний матеріал (обмежену кількість тексту у вигляді заголовків слайдів, плану презентації та переліку задач, функцій, властивостей, компонентів як нумерованих списків; рисунки, таблиці, діаграми тощо.

3) Мова тексту презентації – українська. Дозволяється використання англійської мови за потреби.

4) Рекомендується використовувати шрифт – Arial, розмір шрифту який більше або дорівнює 28 пунктів для тексту презентації.

5) Слайди мають бути пронумеровані починаючи з титульного слайду (номер слайду на титульному слайді не ставиться). Розмір шрифту номеру слайду має бути 28 пунктів.

6) На усну доповідь та презентацію роботи студенту відводиться 5 хвилин. Перевищувати цей час не рекомендується.

7) Рекомендується використовувати наглядні елементи презентації - сформувати та розповісти історію (легенду) роботи тільки у разі, коли ця історія тісно пов’язана з сутністю роботи; в разі потреби можна доводити аналогії, ставити запинання, використовувати цитати та посилання на обгрунтовані дослідження та статистику.

8) Приклад структури презентації представлений у таблиці нижче.

Таблиця - Приклад структури презентації та рекомендована кількість слайдів.

| Обов’язкові елементи презентації | Кількість слайдів |
| --- | --- |
| ТИТУЛЬНИЙ слайд | 1 |
| ВСТУП (короткий опис предметної області та опис актуальної проблеми, вирішення якої планується в роботі, що презентується) | 1-2 |
| Методи та програмний інструментарій, що використовувалися для проекту | 1-2 |
| ПОСТАНОВКА ЗАДАЧІ | 1-5 |
| Програмний інструментарій, що буде використовуватися для реалізації проекту | 1-2 |
| РЕАЛІЗАЦІЯ СИСТЕМИ (основні проектні рішення, діаграми, схеми, скіншоти виконання застосунку тощо) | 8-16 |
| ВИСНОВКИ | 1-2 |
| Кількість слайдів разом | 15-30 |

Порядок підготовки матеріалів у курсовій роботі

При підготовці матеріалів курсової роботи до підшивання у папку потрібно скласти їх у такому порядку та пронумерувати сторінки (нумеруються не всі сторінки – див. список нижче по тексту):

1) титульний лист курсової роботи (не нумерується),

2) індивідуальне завдання на курсову роботу (не нумерується),

3) календарний план (не нумерується),

4) зміст (нумерація сторінок починається з 4 сторінки), аркуш з рамкою із загальними визначеннями для першої сторінки, решта сторінок (нумерація продовжується) – аркуші з рамкою з номером сторінки;

5) реферат (нумерація продовжується);

6) перелік скорочень (нумерація продовжується);

7) вступ (нумерація продовжується);

5) розділ 1 з відповідними пунктами, розміщеними у порядку нумерації (нумерація продовжується);

5) розділ 2 (і наступні, якщо є) з відповідними пунктами, розміщеними у порядку нумерації (нумерація продовжується);

6) висновки (нумерація продовжується);

7) список використаних джерел (нумерація продовжується);

8) додатки у порядку, що визначений їх нумерацією: А, Б, В ... (сторінки  
додатків нумеруються окремо, починаючи з першої, але після номера  
вказується буквене позначення додатку, наприклад: 1А, 2А тощо).

Пояснювальна записка до курсової роботи доповнюється CD-диском, який розміщується у конверті, приклеєному скотчем до внутрішньої сторони обкладинки наприкінці роботи.

# **Додаток 10. Попередній захис**т

Для оцінки якості програмного проекту на цикловій комісії створюються спеціальні комісії і встановлюються терміни захистів проектів і попередній захист курсових проектів. До складу комісії обов'язково повинний бути включений керівник студента. На попередньому захисті, що являє собою репетицію захисту дипломного проекту, комісія розглядає матеріали проекту і слухає доповідь студента. Студент повинний продемонструвати цілком готову пояснювальну записку і графічну частину. На попередньому захисті даються рекомендації з виправлення помилок, і приймається остаточне рішення про допущення або недопущення студента до захисту.

У випадку негативного рішення за результатами попереднього захисту курсового проекту, це питання розглядається в цикловій комісії за участю керівника проекту. Робота вважається виконаною, якщо пояснювальна записка і весь графічний матеріал оформлені відповідно до вимог діючих стандартів. Перевірку правильності здійснює нормоконтролер – один з викладачів циклової комісії і ставить свій підпис на відповідному аркуші пояснювальної записки.

# **Додаток 11. Зразки шаблонів аркушів курсової роботи.**