



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО
ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Слободян Р.О.

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи

з дисципліни

«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Галузь знань: 12 Інформаційні технології

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: «Обслуговування програмних систем і комплексів»

Вид дисципліни: обов'язкова

Відділення телекомунікацій та електронних систем

Циклова комісія комп'ютерних наук

Слободян Р.О. Методичні вказівки до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» для здобувачів фахової передвищої освіти за освітньо-професійною програмою «Обслуговування програмних систем і комплексів» спеціальності 122 «Комп'ютерні науки» галузі знань 12 «Інформаційні технології». «29» серпня 2023 року – 45 с.

Укладач: Слободян Руслан Олесійович, спеціаліст першої категорії.

Методичні вказівки розглянуто і схвалено на засіданні циклової комісії комп'ютерних наук
Протокол №1 від «29» серпня 2023 року

Голова циклової комісії комп'ютерних наук
«29» серпня 2023 року



Галина МАРЦІЯШ

Розглянуто і схвалено методичною радою ВСП «ТФК ТНТУ»
Протокол № 1 від «30» серпня 2023 року

© Слободян Р.О., 2023

© ВСП «ТФК ТНТУ», 2023

ЗМІСТ

ВСТУП.....	4
1 МЕТА ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ	5
2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ	6
2.1 Вибір теми курсової роботи.....	6
2.2 Формулювання вимог до програми.....	7
2.3 Проектування програми	7
2.4 Кодування програми	8
2.5 Тестування програми.....	9
2.6 Оформлення пояснювальної записки	9
2.7 Підготовка до захисту.....	9
2.8 Захист курсової роботи.....	10
3 СКЛАД ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	12
3.1 Структура пояснювальної записки	12
3.2 Зміст і наповнення розділів пояснювальної записки	13
4 ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	25
4.1 Загальні вимоги	25
4.2 Основні написи пояснювальної записки	25
4.3 Структурні елементи та розділи.....	26
4.4 Нумерація сторінок.....	27
4.5 Нумерація розділів та підрозділів	27
4.6 Переліки	28
4.7 Ілюстрації.....	28
4.8 Таблиці	29
4.9 Лістинги	30
4.10 Формули і рівняння.....	32
4.11 Посилання	33
4.12 Скорочення	33
4.13 Перелік посилань	33
4.14 Додатки	34
5 КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ	36
6 СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	38
Додаток А Перелік пропонованих тем курсових робіт	40
Додаток Б Зразок оформлення титульного аркуша	45

ВСТУП

Курсова робота – це перша самостійна науково-дослідницька робота, що виконується здобувачами фахової передвищої освіти протягом семестру з метою закріплення, поглиблення і узагальнення знань, одержаних за час навчання та їх застосування до комплексного вирішення конкретного фахового завдання.

При виконанні курсової роботи студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, освоює методики експериментальних досліджень, аналізу, узагальнення і літературного оформлення одержаних результатів з теми дослідження, набуває вміння вести науковий пошук.

Курсова робота з навчальної дисципліни «Об'єктно-орієнтоване програмування» - це складова частина навчального процесу, яка є завершальним етапом вивчення дисципліни «Об'єктно-орієнтоване програмування». Її написання є обов'язковим елементом підготовки студентів спеціальності 122 «Комп'ютерні науки» та виконується протягом п'ятого семестру.

Дані методичні вказівки дають змогу студентам виконати та оформити курсову роботу відповідно до прийнятих вимог та положень ВСП «ТФК ТНТУ». У методичних вказівках поетапно розписано пункти виконання курсової роботи, подано вимоги до оформлення роботи із необхідними зразками.

1 МЕТА ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ

Курсова робота з навчальної дисципліни «Об'єктно-орієнтоване програмування» передбачає виконання практичного завдання з розробки програмного забезпечення з використанням об'єктно-орієнтованої парадигми програмування. Як правило, завданням курсової роботи є розробка нової або удосконалення існуючої програми, бібліотеки програм, програмного пакета з використанням об'єктно-орієнтованої мови C++ та фреймворку Qt.

Метою курсової роботи є закріплення, поглиблення та узагальнення знань, якими студент оволодів під час вивчення курсу ООП, в набутті навичок використання основ алгоритмізації та програмування на алгоритмічних мовах високого рівня з використанням принципів об'єктно-орієнтованого проєктування та програмування, методики розробки програмних модулів, набуття студентами теоретичних знань та практичних навичок в області використання сучасних систем візуального проєктування та освоєння принципів і методів сучасних технологій програмування, набуття навичок професійних та практики наукових досліджень з оформленням відповідного текстового, програмного та ілюстративного матеріалу у формі наукового звіту.

В результаті вивчення дисципліни та роботи над курсовою роботою студенти повинні:

- знати загальні питання побудови та використання сучасних систем об'єктно-орієнтованого проєктування програмних засобів комп'ютерних систем; основні поняття та методи об'єктно-орієнтованого проєктування та програмування, основні засоби розробки програм з використанням ієрархічних структур об'єктів та пакетів об'єктів;
- вміти самостійно вибирати середовище проєктування програмних засобів, проєктувати, розробляти, налагоджувати та супроводжувати сучасне програмне забезпечення комп'ютерних систем.

Курсова робота є самостійною роботою студента. Відповідальність за правильність аналітичних висновків, результатів проєктування і програмування, а також оформлення несе студент – автор курсової роботи.

2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основні етапи виконання курсової роботи:

- а) вибір теми, отримання індивідуального завдання;
- б) формулювання вимог до програми;
- в) проектування програми;
- г) кодування програми;
- д) тестування програми;
- е) оформлення пояснювальної записки;
- ж) підготовка до захисту;
- з) захист курсової роботи.

Про результати виконання курсової роботи студент обов'язково повинен періодично звітувати перед викладачем - керівником курсової роботи.

2.1 Вибір теми курсової роботи

Теми курсових робіт закріплюються за студентами викладачем особистими побажаннями студентів з урахуванням їх бажання, нахилів і практичного досвіду, із переліку, наведеного у додатку А. За узгодженням з керівником, студент може запропонувати для написання курсової роботи свою тему, яка не увійшла до запропонованого переліку, або дещо змінити назву, обґрунтувавши доцільність.

Перелік закріплених за студентами тем курсових робіт затверджується на засіданні циклової комісії. Після цього, керівник відповідно до теми складає і видає студенту індивідуальне завдання на виконання курсової роботи. Завдання підшивається до курсової роботи, коли її закінчено та разом з нею подається до захисту.

При необхідності студент може отримати кваліфіковану консультацію по отриманій темі у керівника. Тема курсової роботи повинна відповідати інтелектуальним можливостям студента, як з точки зору вибору певної задачі, так і характеру роботи.

2.2 Формулювання вимог до програми

Формулювання вимог є самостійним етапом роботи студента, в якому здійснюється аналіз теми та індивідуального завдання на виконання курсової роботи.

На етапі формулювання вимог студент повинен:

- а) сформулювати призначення програми;
- б) осмислити область застосування;
- в) визначити достатній перелік функцій, які виконуватиме програма;
- г) запропонувати інтерфейс користувача, який надасть можливість останньому використовувати всі функції програми;
- д) сформулювати вимоги до надійності та умови експлуатації;
- е) передбачити порядок контролю та прийому розроблюваного програмного забезпечення.

Результатом виконання даного етапу є перший розділ пояснювальної записки курсової роботи – «Технічне завдання». Детальний огляд необхідних пунктів даного розділу записки наведено в підрозділі 3.2.4.

2.3 Проєктування програми

На етапі проєктування визначається архітектурний стиль, об'єктна структура програми, а також формат вхідних, вихідних та внутрішніх даних. Співвідношення між класами та об'єктами відображується за допомогою діаграм мовою UML. Під час проєктування можна розробити план функціонального тестування та необхідні тести.

Незалежно від теми програма, що розробляється, повинна задовольняти наступним загальним вимогам:

- а) *Стійкість програми.* Програма не повинна втрачати працездатності ні при яких, навіть некоректних, діях користувача. Всякі дії, що загрожують втратою інформації, мають бути підтверджені користувачем. Інформація, що вводиться, скрізь, де це можливо, повинна піддаватись логічному контролю.
- б) *Забезпечення цілісності даних.* При будь-яких діях користувача не

повинні втрачатися дані або їх цілісність (некоректність індексів, втрата посилань в зв'язках після видалення-додавання записів і т. д.).

в) *Функціональна повнота.* Мають бути реалізовані усі функції, вказані в специфікації програми.

г) *Терміни і інтерфейс.* У діалогових вікнах використовуються тільки терміни, зрозумілі користувачеві, і не використовуються терміни розробника («запис», «індексація» і т. д.). Поява службових англomовних повідомлень неприпустимо. У повідомленнях користувача слід дотримуватися норм ввічливості, колірна гамма повинна наслідувати загальноприйняті рекомендації.

д) *Використання клавіатури.* На будь-якому етапі натиснення будь-якої клавіші повинне ігноруватися або викликати передбачені дії, описані в засобах допомоги. Прив'язка дій до клавіш має бути загальноприйнятою: F1 - допомога; Enter - згода, завершення введення; Esc - відмова, повернення до попереднього вузла гілки алгоритму; Tab - перехід до наступного поля, вікна і т. д.; Shift + Tab - повернення до попереднього поля і т., д.

2.4 Кодування програми

На етапі кодування створюється код програми, який повинен повністю відповідати вимогам індивідуального завдання. Код повинен бути гідним чином документований, для зручності його читання необхідно дотримуватись загальноприйнятих норм вирівнювання блоків коду. Якщо в процесі кодування вирівнювання належним чином не виконувалось – необхідно виконати автоматичне вирівнювання (у Visual Studio комбінації клавіш Ctrl+K + Ctrl+D, у Qt Creator – Ctrl+A + Ctrl+I) перед використання коду в пояснювальній записці.

Програмний код має бути наведений у додатку до пояснювальної записки. У тексті основних розділів пояснювальної записки можливо наводити лише невеликі за розміром окремі фрагменти коду.

2.5 Тестування програми

Від правильного виконання даного етапу залежить правильність виконання і стійкість роботи програмного продукту. Розрізняють 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 і виконується у міру того, як кодуються окремі функції програми, а також перед захистом програми.

Модульне тестування робиться одночасно з кодуванням і має за мету оперативну перевірку окремих частин програми. Виконання модульного тестування є вельми бажаним, але не є обов'язковим.

2.6 Оформлення пояснювальної записки

Пояснювальна записка повинна розкривати зміст курсової роботи, містити обґрунтування вибору методів, алгоритмів та програм для вирішення поставленої задачі, аналіз отриманих результатів та інші матеріали. Оформлення пояснювальної записки є важливим етапом виконання курсової роботи і має надати студентові навичок документування програмного продукту. Рекомендований склад пояснювальної записки розглянуто в 3-у розділі, вимоги до оформлення наведені в 4-у розділі.

2.7 Підготовка до захисту

Виконана згідно стандартів у відповідності із завданням і в повному обсязі пояснювальна записка, підписана виконавцем, у незброшурованому вигляді подається на перевірку керівнику.

До пояснювальної записки додається повний електронний варіант всіх файлів з текстами програм та допоміжних програм-утиліт (якщо такі є), тестові файли, скомпільований виконуваний файл основної програми, об'єктні модулі (у випадку розробки модулів або бібліотеки).

Робота подається на перевірку, не пізніше, як за тиждень до захисту.

Виявлені при перевірці курсової роботи неточності і помилки студент зобов'язаний виправити і представити керівникові. Якщо ж при огляді встановлено, що робота в будь-якій частині потребує суттєвої доробки, то визначається обсяг доробки і встановлюється термін подання виправленої роботи на повторну перевірку.

Роботи, що не відповідають затвердженій темі, без затвердженого завдання на курсову роботу, підписаного студентом і викладачем, а також ті, в яких виявлено запозичення з інших джерел, без посилання на джерело, до захисту не допускаються.

2.8 Захист курсової роботи

До захисту курсової роботи студент допускається викладачем – керівником курсової роботи. Необхідними умовами допуску до захисту є наявність:

- а) діючої програми, яка відповідає індивідуальному завданню;
- б) пояснювальної записки у зброшурованому вигляді, складеної і оформленої відповідно до вимог;
- в) правильно оформленого програмного коду, який записаний на оптичний носій інформації і прикріплений до останньої сторінки пояснювальної записки.

Захист відбувається перед комісією з двох викладачів у присутності студентів з академічної групи автора відповідно до встановленого графіку захистів, і проходить в такій послідовності:

- доповідь студента про основні результати виконаної ним роботи, демонстрація програми;
- відповіді студента на запитання присутніх;
- відповіді на зауваження.

Для доповіді про результати виконаної роботи студенту представляється 5-10 хвилин. Доповідь повинна складатися з трьох частин (вступна частина, основна частина, висновки).

У вступній частині доповіді необхідно відмітити актуальність теми, дати загальний аналіз стану питання, сформулювати основні задачі, з рішенням яких зв'язано виконання роботи в цілому.

В основній частині доповіді необхідно привести короткі відомості про зміст виконаної роботи, відмітити основні підходи та показати ефективність прийнятих рішень. Основну частину доповіді можна супроводжувати посиланням на графічні матеріали та демонструванням роботи програми.

У висновках необхідно чітко сформулювати основні результати курсової роботи, наголосивши на повноті виконання поставленої задачі.

Відповіді на питання повинні бути короткі, за суттю і не виходити за межі поставленого питання.

3 СКЛАД ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

3.1 Структура пояснювальної записки

За результатами виконання курсової роботи кожен студент індивідуально і відповідно до завдання складає пояснювальну записку курсової роботи.

Рекомендується наступна структура пояснювальної записки:

Титульний аркуш

Завдання на курсову роботу

Зміст

Вступ

1 Технічне завдання

1.1 Найменування та область застосування

1.2 Підстави для розробки

1.3 Призначення розробки

1.4 Вимоги до програми чи програмного виробу

1.5 Вимоги до програмної документації

1.6 Техніко-економічні показники

1.7 Стадії та етапи розробки

1.8 Порядок контролю та прийому

2 Розробка технічного та робочого проекту

2.1 Розробка загальної структури і варіантів використання програми

2.2 Розробка системи класів

2.3 Розробка методів

2.4 Проектування інтерфейсу користувача

2.5 Опис файлової структури програми

2.6 Опис структури бази даних програми

3 Тестування програми і результати її виконання

Висновки

Перелік посилань

Додаток А Діаграма варіантів використання

Додаток Б Лістинг програми

Додаток В Оптичний диск із програмним продуктом

3.2 Зміст і наповнення розділів пояснювальної записки

3.2.1 Титульний аркуш і завдання на курсову роботу

Титульний аркуш пояснювальної записки курсової роботи оформляється на аркуші формату А4 відповідно до зразку, зображеного в додатку Б. Титульний аркуш має бути підписаний студентом та керівником роботи.

Завдання на курсову роботу є початковим документом для курсової роботи, складається керівником курсової роботи і узгоджується із студентом, що виконує роботу.

Титульний аркуш і завдання на курсову роботу вважаються відповідно першим і другим аркушами пояснювальної записки, однак нумерація сторінок на них не проставляється.

3.2.2 Зміст

Зміст є обов'язковим розділом пояснювальної записки курсової роботи, який включає назву розділів і підрозділів записки (починаючи із вступу), список використаної літератури, додатки і їх найменування, а також номер сторінки, на якій розміщений або починається розділ або додаток.

3.2.3 Вступ

У вступі автор обґрунтовує вибір теми, коротко викладає її актуальність, призначення розробки, для вирішення яких конкретних практичних завдань вона може бути використана. Тут потрібно сформулювати мету і завдання роботи, визначити основні підходи та ідеї, вибрати спосіб розв'язання задач. При наявності аналогів програмних продуктів, які розв'язують подібні задачі, потрібно коротко охарактеризувати основні обмеження та недоліки й запропонувати шляхи їх усунення.

У вступі слід акцентувати увагу на прикладній проблемі, яку треба вирішити в курсовій роботі, а не на засобах її вирішення. Недоцільно у цьому

розділі наводити означення відомих термінів об'єктно-орієнтованого програмування, давати занадто детальні характеристики й описи використаного програмного забезпечення та іншу інформацію, що не стосується теми курсової роботи. Обсяг вступу – 1-2 сторінки.

3.2.4 Технічне завдання

Основною метою даного розділу пояснювальної записки є осмислення і формулювання технічних вимог до програмного продукту, які безпосередньо впливають з теми і завдання на курсову роботу і необхідні для подальшої розробки програмного продукту. Обсяг розділу – 2-4 сторінки

В даному розділі необхідно описати такі підрозділи:

3.2.4.1 Найменування програми та область застосування. Підрозділ повинен містити такі пункти:

- **найменування програми** – подається повне найменування та (або) позначення (стисла назва) програмного вибору;
- **коротка характеристика області застосування** - стисла характеристика області застосування програмного виробу та об'єкта, в якому передбачається його використання.

3.2.4.2 Підстави для розробки. Даний підрозділ повинен містити пункти:

- **підстави для проведення розробки** – зазначаються документи, на підставі яких проводиться розробка, назва організації, що затвердила документи, дата затвердження;
- **найменування і (або) умовне позначення теми роботи.**

3.2.4.3 Призначення розробки – повинно бути викладено експлуатаційне та функціональне призначення програмного виробу:

- **експлуатаційне призначення** – це мета використання майбутнього програмного виробу;
- **функціональне призначення** – це засоби досягнення поставленої мети.

3.2.4.4 Вимоги до програми чи програмного виробу – підрозділ містить перелік основних вимог, реалізація яких дасть змогу розв’язати поставлену задачу. Вимоги мають бути викладені повно, чітко, в термінах, зрозумілих проєктувальнику. Не дозволяється використання формулювання, що мають неоднозначний зміст. До таких вимог відносяться:

- **вимоги до функціональних характеристик** – необхідно вказати необхідний перелік виконуваних програмним продуктом функцій, вимоги до організації вхідних і вихідних даних;

- **вимоги до часових характеристики** – часові обмеження на роботу програми мають бути описані, виходячи з характеру задачі, що розв’язується. Наприклад, вимоги до часових характеристик в інформаційно-пошукових системах можуть становити до 10 сек. на відповідь на запит; в системах керування виробничим процесом - до 1сек., а в програмах, що експлуатуються раз на місяць, жорсткі вимоги можуть не ставитися;

- **вимоги до надійності** – в даному пункті необхідно вказати вимоги до забезпечення надійного функціонування (забезпечення стійкого функціонування, контроль вхідної і вихідної інформації, час відновлення після збою).

- **умови експлуатації** – необхідно описати умови експлуатації програмного продукту (кліматичні умови експлуатації, вимоги до видів обслуговування, вимоги до чисельності і кваліфікації персоналу);

- **вимоги до складу і параметрів технічних засобів** – в даному пункті вказується необхідний склад технічних засобів із зазначенням їх технічних характеристик;

- **вимоги до інформаційної і програмної сумісності** – зазначаються вимоги до вихідних кодів і мови програмування, вимоги до програмних засобів, які буде використовувати програмний виріб, вимоги до захисту інформації і програми.

3.2.4.5 Вимоги до програмної документації - вказується необхідний склад документації за Держстандартом та спеціальні вимоги до неї. Склад програмних документів визначається залежно від складності, призначення

програмного виробу та складу персоналу, що обслуговує програмний виріб.

3.2.4.6 Техніко-економічні показники – необхідно вказати орієнтовну економічну ефективність, приблизне використання за рік.

3.2.4.7 Стадії та етапи розробки – в даному підрозділі визначаються стадії та етапи розробки програмного виробу, зміст їх робіт, терміни виконання і виконавці;

3.2.4.8 Порядок контролю та прийому - вміщує опис умов, за яких буде перевірятися працездатність програми: види і обсяг тестів, вимоги до прийому роботи.

3.2.5 Розробка технічного та робочого проєкту

3.2.5.1 Розробка загальної структури і варіантів використання програми

В даному підрозділі пояснювальної записки необхідно сформулювати загальну структуру програми, змодельовати і описати варіанти використання програмного продукту, а також змодельовати роботу кожного варіанту використання окремо.

Виходячи із поставленого технічного завдання, технічних вимог до програми, потрібно розробити детальний перелік функцій програми – варіантів використання і перелік діючих суб'єктів (користувачів програми, стороннього програмного забезпечення, що впливатиме на роботу). Під час розроблення великого проєкту складно буває визначити, які саме діючі суб'єкти можуть знадобитися. Розробник повинен розглядати кандидатів на ці ролі з погляду їх взаємодії з системою:

- чи вводять вони дані;
- чи очікують приходу інформації від системи;
- чи допомагають іншим діючим суб'єктам.

У більшості ситуацій варіанти використання генеруються діючими суб'єктами, але іноді їх ініціює сама система. Комп'ютерна система може видавати повідомлення у процесі своєї роботи, які не будуть залежати від дій

користувача, наприклад, нагадування про необхідність виконання певної дії.

У цілому все, що повинна робити система, повинно бути описано за допомогою варіантів використання на етапі її розроблення. Це можна відобразити у вигляді UML-діаграми варіантів використання (прецедентів), або, при необхідності, у вигляді блок-схеми (оформленій згідно з вимогами стандартів), чи словесним описом.

За допомогою UML можна будувати інформативні і водночас зрозумілі діаграми варіантів використання. Діючі суб'єкти зображуються чоловічками, варіанти використання – еліпсами. Прямокутна рамка оточує всі варіанти використання, залишаючи за своїми межами діючі суб'єкти. Цей прямокутник називають межею системи. Те, що знаходиться всередині, – програмний продукт, яке розробник намагається створити. На рисунку 3.1 показано діаграму варіантів використання для комп'ютерної системи книжкового магазину.

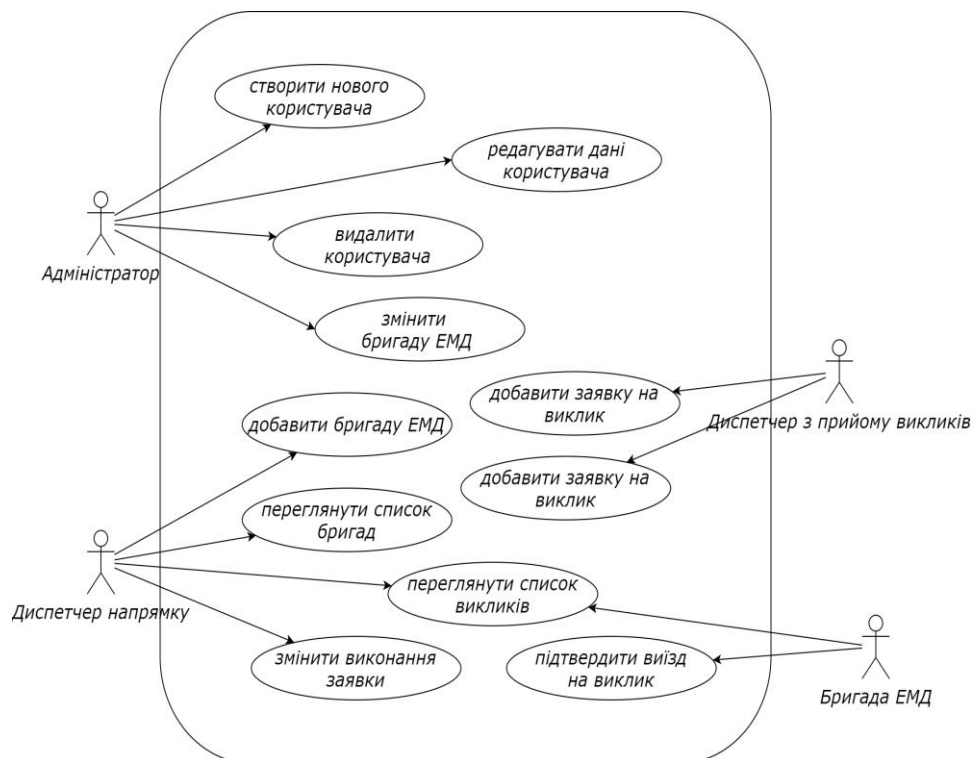


Рисунок 3.1 – Діаграма варіантів використання для книжкового магазину

На цій діаграмі лінії, які називають *асоціаціями*, з'єднують діючі суб'єкти з їх варіантами використання. У загальному випадку асоціації не є

спрямованими, і на лініях немає стрілок, але можна їх вставити для того, щоби наочно показати той діючий суб'єкт, який є ініціатором варіанта використання.

На діаграмі варіантів використання немає місця для розміщення детального опису усіх варіантів використання, тому доводиться виносити описи за її межі. Для створення цих описів можна використовувати різні рівні формалізації, залежно від масштабів проєкту і принципів, якими керуються розробники. У більшості випадків потрібен детальний опис усіх сценаріїв у варіанті використання. Простою реалізацією опису діаграми варіантів використання є один-два абзаци тексту. Іноді використовують таблицю, що складається з двох колонок: діяльність діючого суб'єкта і реакція на неї системи.

Для моделювання роботи варіанту використання також використовують UML-діаграму дій. Цей тип діаграм демонструє керівні потоки від одних дій до інших. Він нагадує блок-схеми, які існували з найперших днів удосконалення технологій програмування. Але діаграми дій повністю формалізовані і мають додаткові можливості. Приклад діаграми дій зображено на рисунку 3.2.

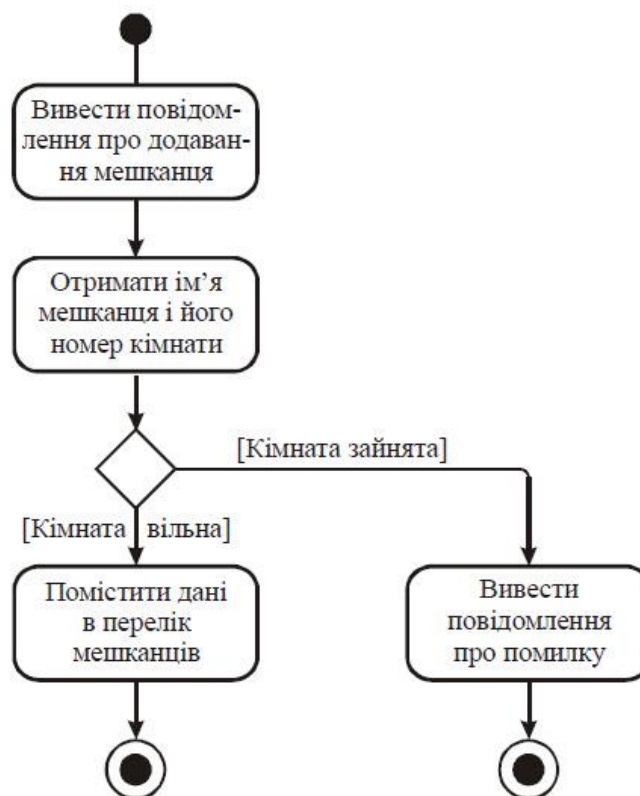


Рисунок 3.2 – UML-діаграма дій

Дії показують на діаграмах ромбоподібними контурами. Лінії, які з'єднують дії, є переходами від одних дій до інших. Розгалуження показане за допомогою ромбів з одним входом і двома або більш виходами. Початковий і кінцевий стани позначаються, відповідно, кружечком і кружечком у кільці.

Діаграми дій можуть також використовуватися для представлення складних алгоритмів, що трапляються в коді програми. В цьому випадку вони практично ідентичні блок-схемам.

Обсяг підрозділу – 3-5 сторінок.

3.2.5.2 Розробка системи класів

В даному підрозділі пояснювальної записки необхідно розробити систему класів і змодельовати її структуру за допомогою мови UML.

Коли визначено всі діючі суб'єкти і варіанти використання, то процес розроблення плавно переходить з етапу проєктування у етап побудови програми. На даному етапі необхідно визначитись із необхідними класами, які входитимуть у програму. Одним з підходів до створення класів є використання імен іменників, що трапляються в коротких описах варіантів використання. Вони є кандидатами в класи, але не з усіх іменників можуть бути повноцінними класами. Потрібно виключити дуже загальні, тривіальні іменники, а також ті, які краще подати у вигляді атрибутів класів.

Після визначення кандидатів у класи можна перейти до визначення методу їх взаємодії. Для цього варто подивитися на дієслова описів варіантів використання. Здебільшого дієслово стає тим повідомленням, яке передається від одного об'єкта до іншого, або передуює тим діям, які виникають між класами.

Доречним буде поділ класів на групи, що мають хоча б щось спільне (дані, властивості, функціональність). В більшості випадків об'єкти (класи) будуть більш чи менш подібні, в групах можна буде виділити підгрупи, і т.д. аж до складного ієрархічного дерева.

В кожній групі виділяється найзагальніші властивості, притаманні всім без винятку об'єктам в групі та дії над ними. Таким чином отримаємо базовий клас для цієї групи об'єктів, даними якого будуть загальні параметри об'єктів групи, а методами - дії, які можна проводити над даними будь-якого з об'єктів

групи. Далі для кожної з підгруп додаємо притаманні їй властивості як поля породженого від неї базового класу, який буде базовим класом для групи. Додаткові дії стають методами цього класу. Породження класів продовжується до тих пір, поки не будуть описані всі параметри об'єктів та дії, які необхідно над ними здійснювати для розв'язку поставленої задачі. Описана послідовність виконується над кожною з груп об'єктів.

Для зображення системи класів в даному розділі пояснювальної записки необхідно використати діаграму класів мови UML (рис. 3.3).

Якщо деякі дії передбачають виконання операцій над кількома об'єктами, і при цьому необхідно мати безпосередній доступ до закритих даних класу, то такі дії робимо дружніми функціями для того класу, над даними якого вони працюють.

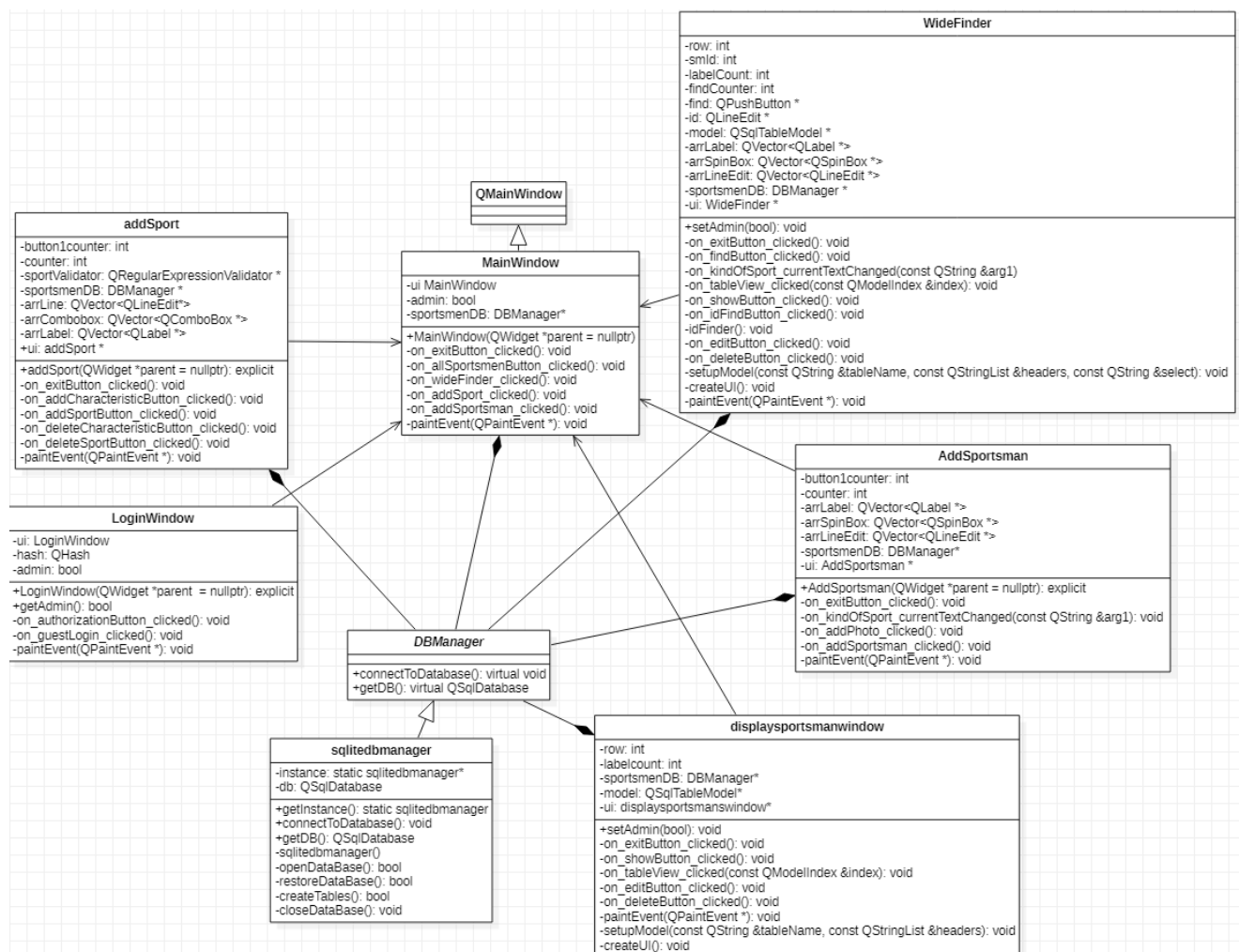


Рисунок 3.3 – UML-діаграма класів

Даний підрозділ відображає основну роботу, яка виконується при об'єктно-орієнтованому програмуванні, і в результаті його виконання отримуються описи всіх класів з даними та методами, причому для методів вже задані їх параметри та тип результату, який вони повертають. Тобто з точки зору інтерфейсу, робота над класами на цьому етапі вже завершена.

Глобальні функції, які не є методами ні одного з класів, також поміщаються в один з модулів, а їх інтерфейсна частина описується в даному підрозділі. Обсяг підрозділу – 2-4 сторінки.

3.2.5.3 Розробка методів

Після того, як з інтерфейсною частиною «закінчено», залишилось лише розробити реалізацію методів, дружніх та глобальних функцій. Ця задача є досить простою, якщо на попередньому етапі було правильно спроектовано систему класів та детально пророблено інтерфейсні частини. Реалізація будь-якої функції (чи методу) розміщується в файлі «*.crr» – частині модуля, на відміну від інтерфейсної, яка записується у «*.h» - файл. Потрібно лише акуратно записати ті дії, які має виконувати метод чи функція, не забуваючи користуватись іншими функціями та методами, замість того щоб весь час змінювати дані за допомогою одних і тих же послідовностей операцій.

У даному підрозділі слід описати принцип дії основних методів, та відмінності поліморфних (віртуальних) методів від методів базових класів.

Обсяг підрозділу – 1-3 сторінки.

3.2.5.4 Проєктування інтерфейсу користувача

Для інтерактивної програми з розвинутою системою меню та діалогових вікон у цьому підрозділі слід описати призначення елементів меню, роботу з ними, параметри, що вибираються в діалогових вікнах тощо. У цьому ж розділі слід описати інтерфейс програм, які працюють з параметрами командного рядка.

Якщо темою роботи є розробка модулів або бібліотеки, то слід детально описати порядок їх використання, навести в алфавітному порядку всі доступні для користувача глобальні типи даних та змінні, функції та класи.

Обсяг підрозділу – 1-2 сторінки.

3.2.5.5 Опис файлової структури програми

Якщо програма використовує файли як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то в даному підрозділі потрібно навести опис формату цих файлів. Він може бути виконаний у текстовому, табличному чи графічному вигляді. Можна додавати до нього приклади реальних файлів з даними програми.

Якщо програма не використовує файли як джерело вхідних даних, то в даному підрозділі необхідно відобразити лише файлову структуру файлів вихідного коду програми, яку необхідно зобразити у вигляді переліку файлів (розширення .h, .cpp, .pro) із поясненням їх призначення, зазначенням додатку даної пояснювальної записки, у якому даний файл наведено.

Обсяг підрозділу – 1-2 сторінки.

3.2.5.6 Опис структури бази даних програми

Якщо програма під час свого виконання використовує локальну або віддалену базу даних для постійного зберігання інформації, а також проміжних чи кінцевих результатів роботи, то в даному підрозділі необхідно детально описати структуру розробленої бази даних.

Даний опис можна представити у вигляді переліку таблиць бази даних із поясненням призначення кожної з них, після чого кожен із таблиць БД потрібно представити у вигляді таблиці, в якій відобразити:

- номер поля по порядку;
- назву поля таблиці бази даних;
- тип поля даних;
- короткий опис даних, що зберігаються в даному полі.

Також доцільно представити структуру розробленої бази даних у вигляді ER-діаграми.

Обсяг підрозділу – 1-2 сторінки.

3.2.6 Тестування програми і результати її виконання

У даному розділі слід описати методику тестування програми, тестові

дані та навести результати роботи програми. Якщо програма працює в графічному режимі, то слід роздрукувати копію графічного вікна програми. Якщо результатом роботи програми є текстовий файл, то необхідно вивести вміст цього файлу. Для програм з розвинутою системою діалогових вікон і меню слід обмежитися друком лише найсуттєвіших результатів, які демонструють правильну роботу програми, а не передруковувати весь екран для кожного відкритого пункту меню. Перелік усіх пунктів меню в такому випадку та вміст неосновних діалогових вікон можна подати в текстовому вигляді.

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

Обсяг розділу – 2-5 сторінок.

3.2.7 Висновки

У висновках по КР в реферативній формі повинні бути описані результати, отримані студентом на кожному з етапів виконання роботи, а також висновків щодо досягнення мети курсової роботи, перспективи впровадження даної програми тощо.

Обсяг – 1-2 сторінки.

3.2.8 Перелік посилань

Тут наводиться перелік джерел, на які були посилання в тексті. В переліку посилань бібліографічні описи подають у порядку, за яким вони вперше згадуються в тексті і вміщують відомості офіційно виданих книжок, статей, патентів, депонованих рукописів тощо.

Обсяг – 1-2 сторінки.

3.2.9 Додатки

У додатки включають лістинги коду, UML-діаграми, додаткові

ілюстрації або таблиці, інші матеріали, які через великий обсяг (більше одного аркуша паперу), специфіку викладення або форму подання не можуть бути внесені до основної частини пояснювальної записки. Також в додатки включають додатковий перелік джерел, на які не було посилянь у тексті записки, але які можуть викликати інтерес.

У додатки необхідно включити повний лістинг розробленого програмного забезпечення. Якщо при розробці використовувалися технології візуального програмування та при значному обсязі коду програми, допускається наводити лише тексти основних модулів програми.

Обов'язковим є включення в останній додаток пояснювальної записки електронної версії розробленого програмного продукту і вихідних кодів, записаних на оптичний носій інформації.

4 ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

4.1 Загальні вимоги

Загальними вимогами до пояснювальної записки є логічна послідовність викладання матеріалу, стислість, чіткість і конкретність викладання теоретичних і практичних результатів роботи, доказовість висновків і обґрунтованість рекомендацій.

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

Пояснювальна записка виконується на одній стороні білого аркушу паперу формату А4 (210 x 297 мм) відповідно до вимог ДСТУ 3008:2015 та ГОСТ 2.105-95, українською мовою із застосуванням друкуючих і графічних пристроїв виведення (ГОСТ 2.004-88).

Усі текстові документи (за винятком титульного аркуша і завдання) повинні мати рамки за формами 9 і 9а ГОСТ 2.106-96. Відстань від рамки до початку і кінця рядків повинна бути не менше 3 мм, а від верхнього або нижнього рядка тексту до лінії рамки не менше 10 мм. Тому доцільно використовувати такі розміри берегів тексту: верхній – 15 мм, нижній – 30 мм, правий – 10 мм, лівий – 25мм.

При оформленні пояснювальної записки дотримуватись наступних параметрів: шрифт – Times New Roman, розмір шрифту – 14 пунктів, колір – чорний, міжрядковий інтервал - 1,5, абзацний відступ першого рядка - 1,5 см.

Обсяг ПЗ повинен складати 25-30 сторінок (без врахування додатків).

4.2 Основні написи пояснювальної записки

Зміст, розміщення та розміри граф основних написів повинні відповідати формам 2 і 2а (ГОСТ 2.104-2006). Для першого аркуша пояснювальної записки із змістом основний напис повинен бути форми 2, а на

інших аркушах змісту і пояснювальної записки – форми 2а. У графах основного напису вказується такі дані:

- у графі 1 – тема курсової роботи, нижче «Пояснювальна записка»;
- у графі 2 – позначення документа. Для навчальних проєктів рекомендується структура позначень зображена на рисунку 4.1;
- у графі 7 – порядковий номер аркуша;
- у графі 8 – загальна кількість аркушів в пояснювальній записці (разом із додатками);
- у графі 9 – вказується індекс навчального закладу (ТК ТНТУ);
- у графі 11 – прізвище та ініціали автора і керівника курсової роботи;
- у графах 12 і 13 – підпис автора роботи і дата підписання відповідно.



Рисунок 4.1 – Структура позначення текстового документу

4.3 Структурні елементи та розділи

Структурні елементи записки, такі як «ЗМІСТ», «ВСТУП», «ВИСНОВКИ», «ПЕРЕЛІК ПОСИЛАНЬ», створюють розділи записки і їхні найменування служать заголовками окремих розділів. Такі розділи не мають нумерації.

Заголовки структурних елементів записки і заголовки розділів необхідно розташовувати посередині рядка і друкувати прописними літерами без крапки наприкінці, не підкреслюючи. Розділи пояснювальної записки можуть

поділятися на підрозділи, пункти та підпункти, які необхідно починати з абзацного відступу і друкувати малими літерами, крім першої прописної, не підкреслюючи і без крапки наприкінці. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

Переноси слів у заголовку розділу не припускаються. Між заголовком і наступним або попереднім текстом необхідно пропустити один пустий рядок або встановити інтервал перед абзацом розміром 24 пт, після –18 пт.

Не припускається розміщувати найменування розділу, підрозділу, а також пункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту.

4.4 Нумерація сторінок

Нумерувати сторінки необхідно арабськими цифрами, дотримуючись наскрізної нумерації по всьому тексті. Нумерацію на титульному аркуші і завданні до курсової роботи не проставляють, аркуш, що слідує за завданням, нумерується цифрою 3. Номер сторінки проставляють у графі 7 (форма 2а ГОСТ 2.104-2006).

4.5 Нумерація розділів та підрозділів

Розділи, підрозділи, пункти, підпункти пояснювальної записки нумеруються арабськими цифрами. Розділи пояснювальної записки повинні мати порядкову нумерацію в межах суті пояснювальної записки і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номери підрозділів складаються з номерів розділів і підрозділів, що розділяються крапкою, наприклад, 1.1, 1.2, 1.3. Номер пункту вміщує номер розділу, підрозділу і пункту, які розділені крапками, наприклад, 3.2.1 – перший пункт другого підрозділу третього розділу.

4.6 Переліки

За необхідністю можуть бути використані переліки. Перед переліком ставлять двокрапку. Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (перший рівень деталізації). Для подальшої деталізації переліку треба використовувати арабські цифри з дужкою (другий рівень деталізації). Переліки першого рівня деталізації виконують з абзацного відступу, другого рівня – з відступом відносно місця розташування переліків першого рівня.

4.7 Ілюстрації

В пояснювальні записці для пояснення тексту, що викладається, повинна бути достатня кількість ілюстрацій – креслень, рисунків, графіків, схем, діаграм, фотознімки чи знімків екрану. Ілюстрації потрібно розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці.

На всі ілюстрації **мають** бути посилання в пояснювальній записці таким чином: «... на рис. 2.3 ...» або «.. на рисунку 3.4....», якщо посилання є повторним – «див. рисунок 1.3». Креслення, рисунки, графіки, схеми, діаграми, розміщені у звіті, мають відповідати вимогам стандартів ЄСКД та ЄСПД.

Ілюстрації повинні мати назву, що розташовують під ілюстрацією. При необхідності під ілюстрацією ще розташовують пояснювальні дані. Ілюстрація позначається словом «Рисунок», що разом із назвою поміщають після пояснювальних даних, наприклад, «Рисунок 2.1 – Схема алгоритму».

Ілюстрації нумеруються арабськими цифрами в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Якщо ілюстрація не вміщується на одній сторінці, можна переносити її на інші сторінки, при цьому назву ілюстрації розташовують на першій сторінці, пояснювальні дані – на кожній сторінці, і під ними вказують: «Рисунок <номер рисунка>, аркуш <номер аркуша>».

4.8 Таблиці

Таблиці необхідно розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі таблиці повинні бути посилання в пояснювальній записці. Таблиця повинна мати назву, яку друкують малими літерами (крім першої прописної) і розміщують над таблицею. Якщо рядки або графи таблиці виходять за формат сторінки, таблицю поділяють на частини, розташовуючи одну частину під іншою, або поруч, або переносячи частини таблиці на наступну сторінку.

Назву таблиці вказують один раз зліва над першою частиною таблиці, над іншими частинами пишуть: «Продовження таблиці <номер таблиці>» з вказівкою номера таблиці. Таблиці слід нумерувати арабськими цифрами в межах розділу, за винятком таблиць, наведених у додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад: «Таблиця 2.3 - Назва таблиці».

У верхній (чи лівій) частині таблиці розміщують заголовок таблиці, в якому вказують назви граф. Діагональний поділ комірок таблиці не допускається. Назви граф пишуть малими буквами, починаючи перше слово графи з великої, у підграфах назви повністю пишуть малими буквами, в кінці крапку не ставлять. Текст в таблицях пишуть з одинарним інтервалом.

Якщо всі параметри величин, які наведені в таблиці, мають одну й ту ж одиницю фізичної величини, то наприкінці назви таблиці після коми розміщують її скорочене позначення (мм). Якщо ж параметри мають різні одиниці фізичних величин, то позначення одиниць записують в заголовках граф після коми (Довжина, мм).

Дані, що наводяться в таблиці, можуть бути словесними і числовими. Слова записують в графах з однієї позиції. Якщо текст складається з одного і більше слів, то при повторенні його замінюють словами “те ж”. При розділенні таблиці горизонтальними лініями - ніякої

заміни не виконують.

Числа записують посередині графі так, щоб їх однакові розряди по всій графі були точно один під одним, за винятком випадку, коли вказують інтервал. Інтервал вказують від меншого числа до більшого з тире між ними: 12 – 35 або 122 – 450. Дробові числа наводять у вигляді десяткових дробів, з однаковою кількістю знаків після коми в одній графі. Якщо цифрові чи інші дані в таблиці не наводяться, то ставиться прочерк.

Якщо таблиця не поміщається на одній сторінці, її переносять на наступну, а під заголовком на початку таблиці нумерують стовпці і цю цифрову нумерацію стовпців розміщують в заголовку продовженої таблиці.

4.9 Лістинги

В тексті пояснювальної записки можуть наводитися ділянки лістингу програми, які доповнюються необхідними коментарями. На кожен лістинг має бути посилання в тексті роботи. Лістинги треба розміщувати безпосередньо після тексту, в якому вони вперше згадуються або на наступній сторінці. Якщо частина лістингу перевищує розмір сторінки, такі ділянки коду розміщуються у додатках.

Лістинги повинні мати порядкову нумерацію в межах кожного розділу. Номер лістингу повинен складатись з номеру розділу та порядкового номеру лістингу, розділених крапкою, наприклад «Лістинг 3.2» - другий лістинг третього розділу.

Лістинги, розміщені в додатках, нумерують у межах кожного додатка, наприклад: «Лістинг А.1.2» – другий лістинг першого розділу додатку А.

При посиланні на лістинг необхідно писати слово «лістинг» із зазначенням його номеру, наприклад «... в лістингу 2.1 ...».

При оформленні лістингів слід використовувати шрифт «Consolas» (або інші моноширинні шрифти «Deja Vu Sans Mono», «Inconsolata», «Courier New»), розмір – 12 пт, міжрядковий інтервал – одинарний. Назва лістингу друкується тим же шрифтом, що і основний текст пояснювальної записки, та розміщується

над лістингом зліва з абзацного відступу через тире після номеру лістингу. Крапка після назви не ставиться.

При оформленні кодів програм, щоб зробити їх зрозумілими та такими, що легко читаються, рекомендується дотримуватись наступних правил:

а) Наявність горизонтальних відступів в тексті програми для відокремлення кожного блоку та вкладених в нього блоків. Відступи робляться за допомогою табуляції (клавіша TAB). За стандартом відступ дорівнює 4 пробілам.

б) Наявність вертикальних відступів, якими для зрозумілості розділяються логічні блоки. До таких логічних блоків можна, наприклад, віднести блок підключення заголовочних файлів, блок об'явлення змінних, блоки операторів вибору та циклів, функції и т. ін. Зазвичай один вертикальний відступ – це один пустий рядок.

в) Супроводження програми коментарями. Їх треба розміщувати в тих місцях програми, де виконуються якісь складні дії. Також коментарями слід починати кожен програму або функцію, для того щоб пояснити, які дії вони виконують. Але коментарі не повинні перефразовувати текст програми, їх призначення - пояснювати мету використання групи операторів, а не описувати дії, що виконують ці оператори.

г) При роботі з консоллю необхідно обов'язково виводити повідомлення (підказку) для користувача про те, які дії він повинен виконати, і про те, що виводить програма.

д) Наявність пробілів у коді. Зазвичай їх роблять довільно для покращання «читабельності» програм. За стандартом пробіл повинен бути після коми, крапки з комою, між логічними та арифметичними блоками, наприклад, до та після операцій +, -, =, >, <= та ін.

е) Розміщення в одному рядку одного оператора. Хоч мова C++ дозволяє використання в одному рядку декількох операторів, це недоцільно робити з таких причин: погіршується «читабельність» програми та виникають труднощі в визначенні синтаксичних помилок, оскільки повідомлення про синтаксичну помилку вказує на номер рядка. Також в цьому випадку незручно

проводити відладку програми, зокрема покрокове виконання програми, здійснювати контроль за зміною значень змінних та ін.

ж) Правильний вибір імен у програмі. Всі об'єкти програми (змінні, функції, типи, іменовані константи і т. ін.) повинні мати обгрунтовані імена, що говорять про призначення того чи іншого об'єкту, наприклад для позначення лічильника в програмі можна використати ім'я counter або increment. Не рекомендується іменувати об'єкти українським транслітом. Хорошим тоном вважається використання англійських слів, наприклад краще використовувати імена maximum або max, а не maksimum. Якщо є обмеження на довжину імені, слід використовувати імена належної довжини. Не слід вживати однакові на вигляд імена, наприклад A10 та AIO.

4.10 Формули і рівняння

Формули і рівняння розташовують у пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, посередині сторінки. Вище і нижче кожної формули повинно бути залишено не менше одного вільного рядка.

Формули і рівняння слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком формул і рівнянь, наведених у додатках. Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, розділених крапкою. Номер формули або рівняння вказують на рівні формули або рівняння в дужках у крайньому правому положенні на рядку.

Пояснення символів і числові коефіцієнти, що входять до формули або рівняння, варто робити безпосередньо після формули або рівняння в тій послідовності, у якій вони надані у формулі або рівнянні. Пояснення значення кожного символу або числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом «де» без двокрапки.

Переносити формули або рівняння на інший рядок припускається тільки на знаках виконуваних операцій, причому знак операції на початку такого

рядка повторюють. При переносі формули або рівняння на знаку операції множення застосовують знак «х». Формули, що прямують одна за одною, і не розділені текстом, відокремлюють комою.

4.11 Посилання

Посилання в тексті на джерела необхідно наводити в послідовності їх згадування в записці, вказуючи порядковий номер, виділений двома квадратними дужками, наприклад, « ... згідно [4 - 7]...». Посилання може містити номер тому (якщо він є) і в необхідних випадках номер сторінки, наприклад: [10, т.2, с.85], [2, с. 37].

При посиланні на стандарти і технічні умови вказують лише їх позначення, наприклад, ГОСТ 2.105-95.

4.12 Скорочення

У тексті пояснювальної записки припускається робити тільки загальноживані скорочення (наприклад «і т.д., і т.п., та ін.») або поширені аббревіатури. Якщо в пояснювальній записці прийнята специфічна термінологія, або вживаються малопоширені скорочення, нові символи, позначення і ін., їх перелік необхідно надати в окремому списку. Перелік повинен розташовуватися стовпчиком, у якому слова за абеткою наводяться скорочено, а справа - їхнє докладне тлумачення. Якщо спеціальні символи, терміни, скорочення, і позначення наводяться менше трьох разів, перелік їх не складається, а їхнє тлумачення роблять у тексті при першому згадуванні.

4.13 Перелік посилань

Перелік використаної літератури повинен містити лише ті літературні джерела, що використані при виконанні курсової роботи і на які є посилання в тексті документу.

Перелік посилань оформлюється як розділ пояснювальної записки з

нової сторінки відповідно до вимог ДСТУ 8302:2015.

Всі джерела нумерують наскрізно арабськими цифрами в порядку їх використання у тексті пояснювальної записки. Про кожен документ подаються такі відомості:

а) Якщо автор один, два або три: прізвища та ініціали авторів через кому перераховуються в порядку, в якому вони вказані в книжці, назва книжки, крапка, місто видання книжки, двокрапка із пробілами з обох боків, назва видавництва, кома, рік випуску, крапка, кількість сторінок, крапка.

Приклади:

Грицюк Ю.І., Рак Т.Є. Об'єктно-орієнтоване програмування мовою С++ : навч. посібник. Львів : Вид-во Львівського ДУ БЖД, 2011. 404 с.

Erdmann K., Schafer W., Mundhenke E. Regierungsorganisation und Verwaltungsaufbau. Heidelberg : D.v. Decker's Verl., 1996. 114 p.

б) Якщо чотири автора: назва книжки, коса риска, прізвища та ініціали авторів через кому перераховуються в порядку, в якому вони вказані в книжці, місто видання книжки, двокрапка із пробілами з обох боків, назва видавництва, кома, рік випуску, крапка, кількість сторінок, крапка.

Приклад:

Ремонт машин / О.І. Сідашенко, О.А. Науменко, А.Я. Поліський та Лубківський Р.М. К. : Урожай, 1994. 400 с.

Бібліографічний опис роблять мовою документа.

Перелік посилань на електронні ресурси потрібно оформляти згідно з наступним зразком:

Статистичні дані ЄСПЛ Analysis of statistics 2016 : вебсайт. URL: http://www.echr.coe.int/Documents/Stats_analysis_2016_ENG.pdf (дата звернення: 01.02.2017).

4.14 Додатки

Додатки необхідно оформляти як продовження пояснювальної записки на наступних її сторінках, або в окремій частині, розташовуючи додатки в

порядку появи посилань на них у тексті пояснювальної записки.

Кожний додаток повинен починатися з нової сторінки і мати заголовок, надрукований вгорі малими літерами з першої прописної, симетрично щодо тексту сторінки. Посередині рядка над заголовком малими літерами з першої прописної повинно бути надруковано слово «Додаток» і прописна буква, що позначає додаток. Додатки слід позначати послідовно прописними буквами українського алфавіту, за винятком букв Г, Є, З, І, Ї, Й, О, Ч, Ь. Один додаток позначається як додаток А.

Додатки повинні мати загальну з іншою частиною пояснювальної записки наскрізну нумерацію. Текст кожного додатка, при необхідності, може бути розділений на підрозділи і пункти, що нумеруються арабськими цифрами в межах кожного додатка. Всі додатки повинні бути перелічені у змісті роботи із зазначенням їх позначень та заголовків.

Ілюстрації, таблиці, формули та рівняння, що є у тексті додатка, слід нумерувати в межах кожного додатка, наприклад, рисунок Г.3 – третій рисунок додатка Г; таблиця А.2 – друга таблиця додатка А; формула (А.1) – перша формула додатка А.

5 КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ

При визначенні оцінки курсової роботи приймається до уваги наступне:

- а) якість програмного продукту;
- б) якість коду;
- в) захист курсової роботи;
- г) якість пояснювальної записки.

Оцінювання курсової роботи проводиться за чотирибальною шкалою: 5 – «відмінно», 4 – «добре», 3 – «задовільно», 2 – «незадовільно».

Оцінка «5» ставиться у випадку, якщо задовольняються всі перераховані нижче вимоги:

- а) в роботі немає суттєвих недоліків;
- б) програма повністю вирішує поставлену задачу;
- в) при розробці програми в повній мірі використано переваги технології об'єктно-орієнтованого програмування;
- г) при захисті роботи студент аргументовано виклав основні технічні рішення, прийняті в процесі розробки, та відповів на поставлені запитання;
- д) пояснювальна записка виконана відповідно до поставлених вимог.

Робота оцінюється на «4», якщо:

- а) в роботі немає суттєвих недоліків;
- б) програма повністю вирішує поставлену задачу, але містить деякі незначні помилки, або при розробці програми не використано всі переваги технології об'єктно-орієнтованого програмування, або програма реалізована не оптимальним чином;

в) при захисті роботи студентом були допущені певні неточності, або не було аргументованих відповідей на деякі з поставлених запитань.

- г) пояснювальна записка виконана відповідно до поставлених вимог.

Робота оцінюється на «3», якщо:

- а) в роботі є суттєві недоліки;
- б) програма не повністю вирішує поставлену задачу, містить значні

помилки, які не дозволяють використовувати її для деяких комбінацій вхідних параметрів;

в) або при розробці програми не використано об'єктно-орієнтованого програмування;

г) робота оформлена зі значними відхиленнями від стандартів та вимог, або в процесі проєктування були відхилення від затвердженого завдання;

д) при захисті роботи студентом допущені суттєві неточності, або не було аргументованих відповідей на поставлені запитання.

Робота оцінюється на оцінку «2», а подальша процедура захисту роботи визначається чинними правилами коледжу, якщо:

а) програма не виконує поставленої задачі;

б) пояснювальна записка не оформлена належними чином;

в) не виконано більшу частину завдання;

г) програма містить значні недоліки, а наявні помилки не дають можливості встановити її працездатність;

д) якщо робота виконана студентом не самостійно, при її захисті не було обґрунтовано прийняті рішення, а задані питання залишились без відповіді.

6 СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Грицюк Ю.І. Рак Т.Є. Об'єктно-орієнтоване програмування мовою C++ : навч. Посібник. Львів : Вид-во Львівського ДУ БЖД, 2011. 404 с.
2. Бублик В.В. Об'єктно-орієнтоване програмування. Київ: ІТ-книга, 2015. 624 с. URL: https://itknyga.com.ua/documents/OOP_final.pdf (дата звернення: 25.08.2023).
3. Алхімова С. М. Навчальний посібник «Об'єктно-орієнтоване програмування». Частина 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення. Київ: КПІ ім. Ігоря Сікорського, вид-во «Політехніка», 2019. 192 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/5c575f1f-a41b-4479-929a-963650ac608d/content> (дата звернення: 25.08.2023).
4. Литвин В.В., Шаховська Н.Б. Проектування інформаційних систем. Львів: Магнолія, 2016. 380 с.
5. Мартін Р. Чистий код. Створення і рефакторинг за допомогою Agile. Харків: Фабула, 2019. 448 с.
6. Швець О. Занурення в патерни проектування. Refactoring Guru, 2021. 395 с.
7. Будай А. Дизайн-патерни – просто, як двері. 2012. 90 с. URL: <https://sites.google.com/site/designpatternseasy> (дата звернення: 25.08.2023).
8. Грицюк Ю.І., Рак Т.Є. Об'єктно-орієнтоване програмування мовою C++. Львів: Вид-во Львівського ДУ БЖД, 2011. 404 с. URL: <https://www.calameo.com/books/007338846547c6c8194ef> (дата звернення: 25.08.2023).
9. Weisfeld M. The Object-Oriented Thought Process, 5th Edition. Addison-Wesley Professional, 2019. 240 p.
10. Stroustrup B. A Tour of C++. 2nd Edition. Addison-Wesley Professional, 2018. 256 p.
11. Соепп В. A Guide to Qt6. Learn to Build Fun Apps & Games for Mobile & Desktop in C++. Apress, 2021. 357 p.
12. Prata S. C++ Primer Plus (Developer's Library) 6th Edition. Addison-

Wesley Professional, 2011. 1140 p.

13. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 1st Edition. O'Reilly, 2014. 334 p.

14. Lafore R. Object-Oriented Programming in C++, 4th Edition. Sams, 2001.

15. Meyers S. Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library. Addison-Wesley Professional, 2001. 288 p.

16. Exceptional C++ Style: 40 New Engineering Puzzles, Programming Problems, and Solutions. Addison-Wesley Professional, 2016. 352 p.

17. Thelin J. Foundations of Qt Development (Expert's Voice in Open Source), 1st ed. Apress, 2007. 550 p.

18. Blanchette J., Summerfield M. C++ GUI Programming with Qt 4, 2nd ed. Prentice Hall, 2008. 752 p.

19. Дистанційний курс «Об'єктно-орієнтоване програмування» на сервері електронного навчання ВСП «ТФК ТНТУ». URL: <https://eguru1.tk.te.ua/course/view.php?id=829>.

20. RuslanSlobodian_HrytsiukRak_ Code examples : вебсайт. URL: <https://github.com/RuslanSlobodian/HrytsiukRak>.

21. Qt Documentation _ Home : вебсайт. URL: <https://doc.qt.io>.

22. Патерни/шаблони проектування : вебсайт. URL: <https://refactoring.guru/uk/design-patterns>.

23. Learn Git Branching : вебсайт. URL: <https://learngitbranching.js.org/?locale=uk>.

24. Уроки програмування на C++ для початківців _ aCode : вебсайт. URL: <https://acode.com.ua/uroki-po-cpp>.

25. C++ ООП – YouTube : вебсайт. URL: https://www.youtube.com/playlist?list=PLHK-VBs5Sk3od4F14xy63llkHyoYxJu_1.

26. C++ Теорія – YouTube : вебсайт. URL: <https://www.youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g>.

Додаток А

Перелік пропонованих тем курсових робіт

а) Бази даних:

- 1) Картотека домашньої бібліотеки.
- 2) Біржа праці (база вакансій та безробітних).
- 3) Записна книжка (база анкетних даних).
- 4) Каса аеропорту (база розкладу, вільних місць, рейсів).
- 5) Квітковий магазин (база квітів, обліку букетів, ціни та інше).
- 6) Аптека (база медикаментів, ліків та приладів, облік, ціни та інше).
- 7) Довідник споживача (служба побуту).
- 8) Довідник покупця (бази торгових точок міста).
- 9) Відділ кадрів (база даних про співробітників фірми).
- 10) Генеалогічне дерево родини.
- 11) Склад (база товарів).
- 12) Каса автовокзалу (база розкладу, вільних місць, рейсів).
- 13) Адміністратор готелю (база номерів, гостей, вільних місць).
- 14) Довідник меломана (база груп і виконавців).
- 15) Щоденник (база намічуваних заходів).
- 16) Термінологія (база визначень будь-якої науки).
- 17) Шеф-кухар (база рецептур страв).
- 18) Довідник лікаря (база хвороб).
- 19) Довідник фірм (назва, адреса, телефони, тощо).
- 20) Обмін житла (база пропозицій по обміну).
- 21) Картотека Інтерполу (база по кожному зареєстрованому злочинцю).
- 22) Бюро знайомств (база потенційних наречених).
- 23) Довідник абітурієнта (база вузів, найменування, адреса, перелік спеціальностей, конкурс минулого року з кожної спеціальності).
- 24) Зарахування абітурієнтів (база абітурієнтів, анкетні дані, сукупність оцінок на вступних іспитах, тощо).
- 25) Довідник поштової індексації (область, район, населений пункт,

поштовий індекс).

26) Ощадбанк (відомості про вкладників банку).

27) Ломбард (база збережених товарів та нерухомості).

28) Довідник селекціонера (найменування сорту будь-якої культури, автор, батьківські сорти, врожайність, характеристики плодів, морозостійкість, стійкість до шкідників і хвороб).

29) Довідник власника відеотеки (база відеофільмів).

30) Купи-продай (база продавців, найменування товару, обсяг партії при оптовому продажі, ціна, умови продажу).

31) Довідник фаната (база спортсменів).

32) Довідник радіоаматора (бази довідкових даних транзисторів, діодів, тиристорів і так далі).

33) Довідник комерційних банків (найменування, адресу, статус (форма власності), умови зберігання коштів, річні відсотки та інше).

34) Довідник начальника в'язниці (анкетні дані ув'язнених, стаття, термін, дата взяття під варту, камера, відомості про родичів, особливості характеру).

35) Довідник командира (список підлеглих військовослужбовців: анкетні дані, адресу батьків, цивільна професія, освіта, звання та дата його отримання, посада, підрозділ, форма служби, особливості характеру і ставлення до служби).

36) Ріелторська контора (база пропозицій купівлі-продажу житла).

37) Довідник туриста (туристичні агенції та пропоновані послуги: країна, місто (або маршрут круїзу), умови проживання та проїзду, екскурсійне обслуговування, сервіс прийнятої боку, вартість путівки).

38) Крилаті фрази (довідник прислів'їв, приказок, афоризмів, каламбурів, інших словесних курйозів).

39) Каталог запчастин автомобіля.

40) Довідник нумізмата (монети: країна, номінал, рік випуску, кількість випущених монет, особливості. Колекціонери: країна, ім'я, контактні координати, наявність рідкісних монет в колекції).

41) Довідник філателіста (марки: країна, національна вартість, рік випуску,

тираж, особливості. Філателісти: країна, ім'я, контактні координати, наявність рідкісних марок в колекції).

42) Довідник любителя живопису (художники з анкетними даними та стилями. Картини з посиланням на художників, датою створення, жанром).

б) Ігри:

43) Гра «Хід конем».

44) Гра «Решітка».

45) Гра «Діагоналі».

46) Гра «Сенегальські шашки».

47) Гра «Японські шашки».

48) Гра «Ланцюг».

49) Гра «Точки».

50) Гра «Мозаїка».

51) Гра «Судоку».

52) Гра «Морський бій».

53) Гра «Словник».

54) Гра «Водопровідник»

55) Гра «Тетріс».

56) Гра «Рикошет».

57) Гра «Сапер».

58) Гра «Зворотній тетріс».

59) Гра «Шашки».

60) Гра «Шахи».

61) Програма для створення і розв'язання японських кросвордів.

62) Програма для створення класичних кросвордів.

в) Освітня діяльність:

63) Інформаційна система навчального закладу.

64) Система тестування.

65) Автоматизація деканату.

66) Розклад занять.

67) Система обліку і аналізу успішності студентів (електронний деканат).

г) Математичні:

- 68) Операції над комплексними числами.
- 69) Обчислення похідних.
- 70) Визначення межі функції.
- 71) Наближене рішення нелінійних і трансцендентних рівнянь.
- 72) Інтерполяція вихідних табличних даних методом Лагранжа.
- 73) Інтерполяція вихідних табличних даних методом Ньютона.
- 74) Обчислення визначених інтегралів методом Монте-Карло.
- 75) Наближене обчислення кратних інтегралів.
- 76) Програмування обчислення площ плоских фігур.
- 77) Пошук екстремумів функції однієї змінної.
- 78) Програмування булевої алгебри.
- 79) Рішення диференціальних рівнянь та їх систем.
- 80) Програмування матричних операцій.
- 81) Обчислення коренів многочленів.
- 82) Програма виконання арифметичних дій над двійковими, вісімковими і шістнадцятковими числами.

д) Захист інформації:

- 83) Алгоритми криптографії: шифр Фейштеля і DES.
- 84) Алгоритми криптографії: шифр Цезаря та шифр Віжинера.
- 85) Алгоритми криптографії: перестановка «Магічний квадрат» та квадрат Полібія.
- 86) Алгоритми криптографії: ГОСТ 28147-89 та DEAL.
- 87) Алгоритми криптографії: Lucifer та MARS.
- 88) Алгоритми криптографії: TEA та Camellia.
- 89) Шифратор/дешифратор тексту з можливістю використання симетричного і асиметричного алгоритмів шифрування.

е) Редактори:

- 90) Текстовий редактор.
- 91) Графічний редактор.
- 92) Музикальний редактор.

ж) Автоматизація:

- 93) Моделювання роботи ліфтів у багатоповерхових будинках.
- 94) Моделювання роботи світлофорів на перехрестях вулиць.
- 95) Ортодоксальний файловий менеджер.

Додаток Б

Зразок оформлення титульного аркуша

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

ВІДДІЛЕННЯ ТЕЛЕКОМУНІКАЦІЙ ТА ЕЛЕКТРОННИХ СИСТЕМ

Циклова комісія комп'ютерних наук

КУРСОВА РОБОТА

з дисципліни:

«Об'єктно-орієнтоване програмування»

на тему: «Тема курсової роботи»

Студента 3 курсу групи КН-321
спеціальності 122 «Комп'ютерні науки»

(ім'я та прізвище)

Керівник: викладач Слободян Р.О.

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії: _____ Руслан СЛОБОДЯН
(підпис)

(підпис)

(ім'я та прізвище)

Тернопіль – 2023