Міністерство освіти і науки України

ВСП «Одеський фаховий коледж комп'ютерних технологій
Одеського національного університету імені І. І. Мечникова»

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ

ОСНОВИ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ

ГАЛУЗЬ ЗНАНЬ: *12 «Інформаційні технології»*
(шифр і назва напряму підготовки)

ШИФР І НАЗВА СПЕЦІАЛЬНОСТІ: *121 «Інженерія програмного забезпечення»*
(шифр і назва спеціальності)

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА: *«Інженерія програмного забезпечення»*
(назва освітньо-професійної програми)

Одеса – 2025 рік

Розробники:
Костянтин МАМУКА, викладач програмування, спеціаліст вищої категорії, аспірант
(вказати авторів, їхні посади, звання, категорії)

Обговорено та схвалено на засіданні циклової комісії
Програмної інженерії та комп'ютерних систем та мереж
(назва циклової комісії)

Протокол № 2 від "26" вересня 2025 року

Голова циклової комісії _____ Костянтин МАМУКА
(підпис)

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ 4
2 ВИБІР ТА ЗАТВЕРДЖЕННЯ ТЕМИ 6
3 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ 7
3.1 Аналіз та постановка задачі 7
3.2 Проєктування програми 7
3.3 Написання програми 7
3.4 Тестування та налагодження 7
4 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ 8
5 АКАДЕМІЧНА ДОБРОЧЕСНІСТЬ ТА ПЕРЕВІРКА НА ПЛАГІАТ 11
6 ЗМІСТ ОСНОВНОЇ ЧАСТИНИ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ 12
7 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОЇ РОБОТИ 14
7.1. Хід виконання та захисту КР 14
7.2. Критерії оцінювання КР 14
8 ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 16
ДОДАТОК А. Перелік завдань
ДОДАТОК Б. Зразок титульного листа
ДОДАТОК В. Зразок завдання на курсову роботу

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Виконання курсової роботи (КР) з освітнього компонента «Основи програмування та алгоритмічні мови» є обов'язковою складовою навчального плану підготовки фахових молодших бакалаврів за освітньо-професійною програмою «Інженерія програмного забезпечення».

Згідно з вимогами освітньо-професійної програми на робочої програми освітнього компонента «Основи програмування та алгоритмічні мови» в процесі виконання курсової здобувачі передвищої фахової освіти повинні набути наступні програмні компетентності та результати навчання:

– загальні компетентності:

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК07. Здатність застосовувати знання у практичних ситуаціях.

– спеціальні (фахові) компетентності:

СК01. Здатність алгоритмічно та логічно мислити.

СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, те-стування, впровадження та супроводу програмного забезпечення.

СК07. Здатність розробляти модулі і компоненти програмного забезпечення за до-помогою типових алгоритмів та інструментів.

СК09. Здатність вибирати та використовувати ефективні інструментальні засоби ро-зробки програмного продукту.

– програмні результати навчання:

РН05. Розробляти та супроводжувати програмне забезпечення.

РН11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.

РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.

Головна мета КР полягає у закріпленні, поглибленні та узагальненні базових теоретичних знань, якими здобувач освіти оволодів під час вивчення розділів «Розділ I. Вступ до програмування мовою С++» та «Розділ II. Основи програмування мовою С++» освітнього компонента «Основи програмування та алгоритмічні мови», їх застосуванні до комплексного вирішення конкретного фахового завдання. Такий підхід повністю відповідає концепції формування висококваліфікованих фахівців у галузі інформаційних технологій, котрі набувають не тільки знань, але й навичок та вмінь, якими повинні володіти випускники закладу фахової передвищої освіти.

Основними цілями написання курсової роботи є:

– закріплення, поглиблення та узагальнення знань, якими здобувач освіти оволодів під час вивчення освітнього компонента;

– надбання досвіду роботи з літературними та фондовими матеріалами, вміння узагальнювати та аналізувати наукову інформацію, виробляти власне ставлення до наукової чи практичної проблеми;

– набуття навичок використання основ алгоритмізації та програмування на алгоритмічних мовах високого рівня з використанням принципів процедурно-орієнтованого програмування;

– набуття здобувачами освіти теоретичних знань та практичних навичок в області використання сучасних систем створення програмного забезпечення та освоєння принципів та методів сучасних технологій програмування;

– вироблення вміння застосовувати методи обчислювальної математики та прикладного програмування для розв’язання прикладних задач;

– проведення ґрунтовного аналізу отриманих результатів і формування змістовних висновків стосовно їх якості.

Під час виконання курсової роботи здобувач освіти повинен продемонструвати:

– вміння збирати і аналізувати відповідні матеріали про об’єкт дослідження, використовуючи сучасні джерела інформації, включаючи Інтернет-ресурси;

– спроможність проводити необхідні обґрунтування для розробки програмного забезпечення різного призначення, тощо;

– здатність доводити розв’язання поставленої задачі до логічного завершення;

– вміння аналізувати отримані результати і робити відповідні висновки.

Курсова робота є самостійною роботою здобуача. Відповідальність за правильність аналітичних висновків, результатів розрахунків і моделювання, а також оформлення несе здобувач освіти – автор КР.

Тема курсової роботи: «Розробка прикладного консольного застосунку».

На виконання курсової роботи відводиться 30 годин (1 кредит ЄКТС) самостійної роботи здобувача освіти.

Після виконання та оформлення КР здобувач освіти захищає її. При цьому береться до уваги компетентність здобувача освіти, оригінальність та творчість мислення, обґрунтованість прийнятих рішень, ритмічність у роботі (дотримання термінів здачі роботи або її складових частин).

2 ВИБІР ТА ЗАТВЕРДЖЕННЯ ТЕМИ

Здобувачі освіти можуть самостійно обрати завдання роботи із затвердженого цикловою комісією переліку. Також дозволяється пропонувати власне ініціативне завдання, виходячи з особистих інтересів та знань. Таке завдання має бути узгоджене з керівником курсової роботи та затверджене цикловою комісією.

Обране завдання закріплюється за здобувачем після оформлення індивідуального завдання на курсову роботу разом із керівником. У випадку, якщо здобувач без поважної причини не обрав завдання у визначений термін, циклова комісія має право призначити тему на власний розсуд.

Виконання курсових робіт з однаковим завданням різними здобувачами не допускається.

Зміна затвердженого завдання можлива як виняток. Для цього здобувач повинен подати мотивовану заяву на ім'я голови циклової комісії, попередньо узгоджену з керівником, не пізніше ніж за один місяць до планового терміну захисту.

3 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основні етапи виконання курсової роботи:

- аналіз та постановка задачі;
- проєктування програми;
- написання програми (кодування);
- тестування та налагодження;
- оформлення пояснювальної записки;
- захист курсової роботи.

3.1 Аналіз та постановка задачі

На цьому етапі здобувач проводить аналіз індивідуальної теми, визначає мету та основні завдання проєкту, описує ключовий функціонал майбутнього застосунку, а також визначає формати вхідних та вихідних даних.

3.2 Проєктування програми

На цьому етапі розробляється архітектура застосунку, що включає: визначення ключових структур даних (struct), проєктування набору функцій (модулів) та зв'язків між ними, розробку логіки роботи меню та сценаріїв взаємодії з користувачем. Також на цьому етапі відбувається вибір інструментальних засобів розробки (середовище розробки, компілятор, додаткові бібліотеки). Для найскладніших алгоритмічних частин створюються блок-схеми.

3.3 Написання програми

На етапі кодування створюється програмний код мовою C++ згідно зі спроєктованою архітектурою. Лістинги повинні бути структурованими, детально прокоментованими та повністю відповідати розробленим алгоритмам.

3.4 Тестування та налагодження

На цьому етапі складається план тестування, який враховує всі особливості програми, включаючи перевірку всіх функцій, коректності обробки даних та стійкості до некоректного вводу користувача. Тестовий набір необхідно узгодити з керівником. Результати тестування та виявлені обмеження програми необхідно задокументувати.

4 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

4.1. Курсова робота є звітною документацією здобувача освіти у сфері техніки і має відповідати встановленій структурі та правилам оформлення. КР має бути написана державною мовою згідно із Законом України «Про забезпечення функціонування української мови як державної» № 2704-VIII від 25.04.2019. Додатки можуть бути оформлені державною або англійською мовою.

4.2. Текстова частина пояснювальної записки (ПЗ) разом з ілюстраціями виконується на одному боці аркушів білого паперу формату А4 (210×297 мм). У разі потреби для графічної частини можна використовувати аркуші формату А3 (297×420 мм).

4.3. Поля сторінки: ліве – 30 мм, верхнє та нижнє – 15 мм, праве – 10 мм. Текст друкується з використанням редактора Word, шрифт – Times New Roman, колір – чорний, накреслення – пряме, кегль – 14 пт, міжрядковий інтервал – 1,5. Таблиці оформлюються кеглем 12 пт з міжрядковим інтервалом 1,0. Основний текст оформлюється не жирним шрифтом і без курсиву; виділення допускається для заголовків та підзаголовків.

4.4. Протягом усієї ПЗ необхідно дотримуватися рівномірної щільності, контрастності й чіткості тексту та рисунків. Усі лінії, літери, цифри та знаки мають бути однаково чорними, чіткими та нерозпливчастими. Використання коректора або інших засобів для виправлення тексту заборонено.

4.5. У тексті не допускається застосування математичних знаків (>, <, =, ≠), а також знаків № (номер) чи % (відсоток) без числових значень. Перед від'ємними значеннями величин у тексті замість знака «–» слід писати слово «мінус».

4.6. Пояснювальна записка повинна мати таку структуру:

- титульний аркуш;
- завдання на КР;
- зміст;
- перелік скорочень та умовних познак (за необхідністю);
- вступ;
- основна текстова частина, викладена за розділами;
- висновки;
- перелік джерел посилання;
- додатки (за необхідністю).

4.7. Сторінки ПЗ нумеруються арабськими цифрами, дотримуючись наскрізної нумерації. Номер сторінки проставляється у правому верхньому куті аркуша без крапки в кінці. Титульний аркуш, завдання на КР, зміст та перелік скорочень включаються до загальної нумерації, але номери сторінок на них не проставляються.

4.8. Титульний аркуш містить відомості про назву КР, спеціальність, виконавця та керівника. Взірець титульного аркушу наведено у Додатку А.

4.9. Завдання на КР містить вихідні дані, зміст основних розділів та терміни виконання. Типовий бланк завдання подано у Додатку Б.

4.10. Зміст подається на новій сторінці одразу після завдання. До нього включаються всі структурні елементи роботи: «Вступ», назви всіх розділів та підрозділів, «Висновки», «Перелік джерел посилання» та назви всіх додатків із зазначенням початкових номерів сторінок.

4.11. Перелік скорочень та умовних познак (за наявності) подається на новій сторінці

після змісту. Перелік слід розташовувати стовпцем: ліворуч в абетковому порядку наводять скорочення, а праворуч – їх розшифрування.

4.12. У вступі, який починається з нової сторінки, коротко викладається актуальність теми, мета та основні завдання курсової роботи.

4.13. Основна частина поділяється на розділи та підрозділи. Заголовки розділів друкуються великими літерами, жирним шрифтом, вирівнюються по центру. Заголовки підрозділів і пунктів друкуються з абзацного відступу маленькими літерами (крім першої великої), без крапки в кінці. Заголовки підрозділів виділяються жирним шрифтом.

4.14. Переліки всередині пунктів або підпунктів починаються з двокрапки.

- Переліки одного рівня підпорядкованості позначаються тире, текст починається з малої літери і закінчується крапкою з комою.
- Багаторівневі переліки позначаються послідовно: малими літерами з дужкою, арабськими цифрами з дужкою, тире. Текст таких пунктів починається з великої літери.

4.15. Відстань між заголовком розділу і текстом (або назвою підрозділу) має становити один порожній рядок (1,5 інтервали). Не допускається розміщувати заголовок у нижній частині сторінки, якщо після нього залишається менше трьох рядків тексту.

4.16. Відстань між заголовком підрозділу і подальшим текстом має становити один порожній рядок.

4.17. Рисунки та таблиці відокремлюються від основного тексту одним порожнім рядком зверху та знизу.

4.18. Абзацний відступ має бути однаковим упродовж усього тексту і дорівнювати 1,25 см.

4.19. Формули та рівняння розміщуються по центру рядка, одразу після тексту, де вони згадуються. Нумерація формул наскрізна, номер ставиться у круглих дужках у крайньому правому положенні рядка.

4.20. Формули набираються у вбудованому редакторі формул. Пояснення значень символів наводяться безпосередньо під формулою, починаючи зі слова «де» без двокрапки.

4.21. Цифровий матеріал оформлюється у вигляді таблиць. Таблиці нумеруються арабськими цифрами в межах розділу (напр., «Таблиця 2.1» – перша таблиця другого розділу). Номер та назва розміщуються над таблицею з вирівнюванням по лівому краю. Назва пишеться з великої літери через тире після номера. На кожну таблицю має бути посилання в тексті (напр., «... наведено в табл. 2.1»). При переносі таблиці на іншу сторінку над продовженням пишуть «Продовження таблиці 2.1».

4.22. Висновки розміщують, починаючи з нової сторінки. У них підбиваються підсумки виконаної роботи.

4.23. Перелік джерел посилання розміщується на новій сторінці та містить лише ті джерела, на які є посилання в тексті. Посилання в тексті подаються у квадратних дужках із зазначенням номера джерела за списком. Оформлення переліку здійснюється згідно з вимогами Національного стандарту України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».

4.24. Додатки містять допоміжні матеріали (напр., повний лістинг коду). Кожен додаток починається з нової сторінки та позначається великою літерою української абетки (напр., «Додаток А»).

4.25. Рисунки нумеруються арабськими цифрами в межах розділу (напр., «Рисунок 3.1» – перший рисунок третього розділу). Номер та назва розміщуються під рисунком по центру. На кожен рисунок має бути посилання в тексті (напр., «... показано на рис. 3.1»).

4.26. Оформлення лістингів коду. Фрагменти програмного коду, що наводяться в основній частині роботи, слід оформлювати моноширинним шрифтом (напр., Courier New або Consolas) кеглем 10-12 пт з одинарним міжрядковим інтервалом. Повний лістинг програмного коду виноситься в додатки та оформлюється аналогічним чином.

5 АКАДЕМІЧНА ДОБРОЧЕСНІСТЬ ТА ПЕРЕВІРКА НА ПЛАГІАТ

Курсова робота є самостійною роботою здобувача освіти. Дотримання принципів академічної доброчесності є обов’язковою умовою для її успішного виконання та захисту.

Під плагіатом розуміється не лише копіювання текстової частини пояснювальної записки, але й використання чужого програмного коду (повністю або частково) без належного посилання на джерело та глибокого розуміння його роботи.

Дозволяється вивчати та аналізувати приклади коду з навчальних посібників, документації та освітніх ресурсів. Використання невеликих фрагментів коду з відкритих джерел можливе лише за умови чіткого посилання на джерело у пояснювальній записці та в коментарях до коду.

Категорично забороняється: подання чужої роботи як власної; копіювання роботи або її значних частин в інших здобувачів освіти; використання програмних засобів для автоматичної генерації всього проєкту без самостійного написання та розуміння коду.

Усі курсові роботи проходять обов’язкову перевірку на наявність текстових та програмних запозичень. Під час захисту керівник також перевіряє рівень самостійності виконання роботи та глибину розуміння здобувачем освіти представленого програмного коду. Роботи, у яких виявлено порушення академічної доброчесності, до захисту не допускаються та оцінюються незадовільно.

6 ЗМІСТ ОСНОВНОЇ ЧАСТИНИ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Основними документами, що представляють курсову роботу, є пояснювальна записка та вихідний код розробленого застосунку на електронному носії. Текст ПЗ повинен бути викладений лаконічно, в обґрунтованому технічному стилі.

Основна частина ПЗ повинна мати чітку структуру, що відображає етапи розробки програмного продукту. Рекомендована структура включає такі розділи:

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

У цьому розділі необхідно детально описати сутність завдання, визначити мету розробки та основні функціональні вимоги до застосунку.

- Аналіз предметної області: опис проблеми, яку вирішує застосунок.
- Мета та завдання проєкту: чітке формулювання мети роботи та перелік завдань, які необхідно вирішити для її досягнення.
- Вимоги до функціоналу: детальний перелік функцій, які повинна виконувати програма (напр., «додавання нового запису», «фільтрація даних за критерієм», «збереження результату у файл»).
- Вхідні та вихідні дані: опис формату даних, з якими працює програма.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ЗАСТОСУНКУ

Цей розділ присвячений опису архітектури та внутрішньої логіки програми.

- 2.1. Вибір засобів розробки. Обґрунтування вибору мови програмування (C++), середовища розробки (IDE, напр., Visual Studio, Clion), компілятора та допоміжних бібліотек.
- 2.2. Архітектура програми. Опис загальної структури проєкту. Необхідно визначити ключові структури даних (struct), які використовуються для зберігання інформації, та описати призначення основних функцій (модулів) програми та зв’язки між ними.
- 2.3. Опис ключових алгоритмів. У цьому підрозділі викладається логіка роботи найскладніших частин програми. Для візуалізації алгоритмів слід використовувати блок-схеми, створені згідно з чинними стандартами. Також тут наводяться необхідні розрахункові формули та теоретичні відомості.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

Розділ описує процес написання коду та ключові програмні рішення. Не слід наводити в цьому розділі повний лістинг коду.

- Опис структури коду: пояснення, які файли містить проєкт та за що відповідає кожен з них.
- Опис основних функцій та структур: наведення та пояснення найбільш важливих фрагментів коду (ключові структури даних, основний цикл програми, найскладніші функції) з детальними коментарями.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ РОБОТИ

У цьому розділі наводяться докази працездатності розробленого застосунку.

- План тестування: опис тестових випадків (test cases), що покривають основний функціонал програми. Для кожного випадку зазначаються вхідні дані та очікуваний результат.
- Результати тестування: демонстрація роботи програми за допомогою знімків екрана (скріншотів), що ілюструють виконання тестових випадків. Кожен знімок екрана має супроводжуватися детальним описом того, що на ньому зображено.

РОЗДІЛ 5. ІНСТРУКЦІЯ КОРИСТУВАЧА

Цей розділ описує, як користуватися розробленим застосунком.

- Призначення програми: короткий опис того, для чого призначений застосунок.
- Порядок роботи: покрокова інструкція щодо запуску програми та використання її основного функціоналу. Опис має містити знімки екрана з поясненнями елементів інтерфейсу користувача (меню, запити на введення даних тощо).

7 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОЇ РОБОТИ

7.1. Хід виконання та захисту КР

Керівник здійснює контроль за ходом виконання курсової роботи здобувачем освіти, надає йому необхідну консультативну допомогу.

Здобувач освіти демонструє викладачу поточні результати роботи над проєктом згідно з календарним планом.

У терміни, визначені графіком, курсова робота здається керівникові на перевірку. КР перевіряється по суті на відповідність завданню, дотримання вимог до оформлення та академічної доброчесності.

Після перевірки роботи та усунення зауважень керівник призначає день, час і місце захисту.

Напередодні захисту здобувачу освіти необхідно повторити теоретичний матеріал, що стосується роботи, та переглянути безпосередньо її зміст.

Захист КР проводиться у формі співбесіди зі з'ясуванням усіх питань, що виникли у керівника під час перевірки роботи.

За результатами захисту, у відповідності до критеріїв оцінювання, що наведені у підрозділі 5.2, викладач разом з членами комісії виставляють здобувачу освіти оцінку.

На підсумкову оцінку за КР впливають:

- якість розробленого програмного забезпечення;
- якість розробленої пояснювальної записки;
- компетентність та загальна ерудиція здобувача освіти при відповідях на запитання під час захисту.

Якщо здобувач освіти подав на захист не самостійно виконану роботу, про що свідчить його некомпетентність у прийнятих рішеннях та матеріалах роботи, КР до захисту не допускається. Такий самий висновок робиться у випадку, якщо КР не завершена на час захисту. У цих випадках робота оцінюється «незадовільно».

7.2. Критерії оцінювання КР

Оцінка з курсової роботи має дві складові: виконання курсової роботи та її захист. Перша складова характеризує роботу здобувача освіти над проєктом та її результат — якість пояснювальної записки та розробленого програмного забезпечення. Друга складова характеризує якість захисту курсової роботи.

Критерії оцінювання виконання курсової роботи наведено в таблиці 5.1.

Таблиця 5.1 – Узагальнені критерії оцінювання курсової роботи

За 4-бальною шкалою	**Оцінка в ЄКТС**	**Критерії оцінювання**
5 (відмінно)	**A**	Пояснювальна записка: повністю відповідає вимогам, аналіз предметної області виконано глибоко, архітектура програми спроєктована оптимально. Програмний продукт: функціонал реалізовано повністю, код добре структурований, прокоментований та працює без помилок. Захист: здобувач освіти демонструє глибокі знання, вільно володіє матеріалом, чітко та обґрунтовано відповідає на запитання.
4 (добре)	**B**	Пояснювальна записка: в цілому відповідає вимогам, але містить незначні неточності в аналізі чи проєктуванні. Програмний продукт: функціонал реалізовано повністю, але є некритичні зауваження до структури коду або логіки роботи. Захист: здобувач освіти добре володіє матеріалом, але припускається незначних помилок у відповідях.
	C	Пояснювальна записка: містить окремі недоліки у структурі або оформленні, аналіз предметної області недостатньо глибокий. Програмний продукт: основний функціонал реалізовано, але присутні логічні помилки, що не призводять до аварійного завершення. Захист: здобувач освіти орієнтується в матеріалі, але відповіді неповні, є труднощі з поясненням коду.
3 (задовільно)	**D**	Пояснювальна записка: структура порушена, аналіз поверхневий, є помилки в оформленні. Програмний продукт: реалізовано лише базовий функціонал, програма працює нестабільно, є суттєві логічні помилки. Захист: здобувач освіти має лише загальне уявлення про роботу, не може пояснити ключові аспекти проєкту.
	E	Пояснювальна записка: не відповідає значній частині вимог. Програмний продукт: реалізований функціонал не відповідає завданню, результат роботи некоректний. Захист: здобувач освіти частково орієнтується в матеріалі, але не може пояснити власний код.
2 (незадовільно)	**FX**	Пояснювальна записка: неструктурована, теоретичний аналіз відсутній. Програмний продукт: програма не компілюється або не виконує жодної з поставлених задач. Захист: здобувач освіти не володіє матеріалом, не може відповісти на запитання, виявлено значні запозичення без посилань.
	F	Курсову роботу не допущено до захисту через невідповідність вимогам, порушення академічної доброчесності або невиконання роботи у встановлений термін.

8 ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. Київ : ДП «УкрНДНЦ», 2016. 17 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. Київ : ДП «УкрНДНЦ», 2017. 29 с.
3. Трофименко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. С++. Алгоритмізація та програмування : підручник. 2-ге вид., перероб. і допов. Одеса : Фенікс, 2019. 477 с.
4. Рудий Т. В., Паранчук Я. С., Сеник В. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2023. 240 с.
5. Основи програмування на C/C++ в прикладах. Частина 1 : навч.-метод. посібник / М. О. Соболь та ін. Харків : НТУ «ХПІ», 2021. 113 с.
6. Основи програмування на C/C++ в прикладах. Частина 2 : навч.-метод. посібник / М. О. Соболь та ін. Харків : НТУ «ХПІ», 2022. 200 с.
7. Standard C++ Foundation : офіційний ресурс зі стандартів та документації. URL: https://isocpp.org/ (дата звернення: 08.10.2025).
8. Learn C++ : інтерактивний ресурс для вивчення C++. URL: https://www.learn-cpp.com/ (дата звернення: 08.10.2025).

ДОДАТОК А. Перелік завдань

ПЕРЕЛІК ЗАВДАНЬ ДО КУРСОЇ РОБОТИ
НА ТЕМУ
«РОЗРОБКА ПРИКЛАДНОГО КОНСОЛЬНОГО ЗАСТОСУНКУ»
З ОСВІТНЬОЇ КОМПОНЕНТИ
«ОСНОВИ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ»

Обговорено та схвалено на засіданні циклової комісії
Програмної інженерії та комп'ютерних систем та мереж
(назва циклової комісії)

Протокол № 2 від "26" вересня 2025 року

Голова циклової комісії _____ Костянтин МАМУКА
(підпис)

Категорія 1: Утиліти для обробки даних та файлів

№	Завдання	Опис завдання
1	Розробка утиліти для аналізу CSV-файлів.	Створити консольний застосунок, що читає CSV-файл, дозволяє користувачеві виконувати сортування даних за обраною колонкою, фільтрацію за значенням та виводити результат у консоль або зберігати у новий CSV-файл.
2	Розробка консольного аналізатора файлів-логів.	Написати програму, що аналізує текстовий файл логів (напр., з записами формату [YYYY-MM-DD HH:MM:SS] [LEVEL] Message). Програма повинна вміти фільтрувати записи за рівнем (ERROR, WARNING), датою та виводити статистику.
3	Розробка утиліти для пакетного перейменування файлів.	Створити застосунок, що дозволяє перейменовувати файли у вказаній директорії за заданим шаблоном (напр., додати префікс, змінити розширення, додати нумерацію).
4	Розробка простого архіватора на основі алгоритму RLE.	Реалізувати утиліту, що стискає та розпаковує текстові файли за допомогою алгоритму Run-Length Encoding (кодування довжин серій).
5	Розробка консольного менеджера конфігураційних файлів.	Написати програму для роботи з файлами конфігурації формату ключ=значення. Утиліта повинна дозволяти читати, змінювати та додавати нові параметри у файл.
6	Розробка утиліти для шифрування/дешифрування файлів шифром Цезаря.	Створити застосунок, що шифрує та дешифрує вміст текстових файлів за допомогою шифру Цезаря із заданим зсувом.
7	Розробка консольного переглядача JSON-даних.	Написати програму, що читає .json файл, перевіряє його на базову коректність (баланс

		дужок) та виводить його структуру у вигляді дерева.
8	Розробка утиліти для порівняння двох текстових файлів.	Створити застосунок, що порівнює два текстові файли та виводить номери рядків, де вони відрізняються, а також самі рядки, що не збігаються.
9	Розробка генератора тестових даних.	Написати програму, що генерує великий текстовий файл із випадковими даними (ПІБ, дати, числа) за заданими параметрами для тестування інших програм.
10	Розробка утиліти для пошуку дублікатів файлів.	Створити застосунок, що сканує вказану директорію, знаходить файли з однаковим вмістом (порівнюючи їх побайтово) та виводить список дублікатів.
11	Розробка конвертера Markdown у HTML.	Реалізувати утиліту, що читає текстовий файл з базовою Markdown-розміткою (заголовки #, списки *) і конвертує його у простий HTML-файл.
12	Розробка утиліти для аналізу текстового твору.	Написати програму, що аналізує текстовий файл, підраховує кількість слів, речень, знаходить найчастіше вживані слова та виводить статистику.
13	Розробка утиліти "дерево директорій".	Створити застосунок, що рекурсивно обходить вказану директорію та виводить її структуру у вигляді дерева, як це робить команда tree.
14	Розробка утиліти для резервного копіювання.	Написати програму, що копіює файли із заданої директорії до іншої, копіюючи лише нові або змінені файли.
15	Розробка утиліти для роботи з INI-файлами.	Створити застосунок для читання та редагування INI-файлів, що мають секції [Section] та пари ключ=значення.

Категорія 2: Консольні ігри та симулятори

№	Завдання	Опис завдання
16	Розробка консольної гри «Змійка» (Snake).	Реалізувати класичну гру «Змійка», де гравець керує змійкою, що збирає їжу, збільшується в розмірах та не повинна врізатися в стіни або у власний хвіст.
17	Розробка консольної гри «Тетріс».	Створити консольну версію гри «Тетріс» з різними фігурами, можливістю їх обертання та системою підрахунку очок за заповнені лінії.
18	Розробка текстової рольової гри (Text-based RPG).	Спроєктувати гру, де гравець може переміщатися між локаціями, взаємодіяти з персонажами, брати участь у покрокових боях та розвивати свого героя.
19	Розробка консольної гри в жанрі	Створити гру з процедурно генерованими

	"Roguelike".	рівнями, де гравець досліджує підземелля, бореться з монстрами та шукає вихід.
20	Розробка консольної гри «Морський бій».	Реалізувати гру «Морський бій» для двох гравців або гри з комп'ютером, з можливістю розставляти кораблі та вести вогонь по полю супротивника.
21	Розробка консольних перегонів (Console Racing).	Створити гру, де гравець керує автомобілем (символом), що рухається по трасі, ухиляючись від перешкод. Швидкість гри поступово зростає.
22	Розробка симулятора життя (Conway's Game of Life).	Реалізувати клітинний автомат «Життя», де початкова конфігурація клітин еволюціонує за певними правилами на кожному кроці.
23	Розробка консольної гри «Сапер».	Створити консольну версію гри «Сапер», де гравець відкриває клітинки на полі, намагаючись не натрапити на міни.
24	Розробка логічної гри «Бики та корови».	Реалізувати гру, де комп'ютер загадує число, а гравець намагається його вгадати, отримуючи у відповідь кількість «биків» та «корів».
25	Розробка консольної гри «Арканоїд».	Створити гру, де гравець керує платформою, відбиваючи м'ячик, який руйнує блоки у верхній частині екрана.
26	Розробка симулятора банкомату.	Написати програму, що імітує роботу банкомату: запит PIN-коду, перевірка балансу, зняття готівки, поповнення рахунку. Дані про рахунки зберігати у файлі.
27	Розробка текстового квесту (Interactive Fiction).	Створити гру, де гравець читає опис ситуації та обирає дії з запропонованих варіантів, просуваючись по сюжету.
28	Розробка гри-вікторини.	Написати програму, що завантажує питання та варіанти відповідей з файлу, ставить їх гравцеві та веде підрахунок правильних відповідей.
29	Розробка симулятора екосистеми «Хижаки та жертви».	Реалізувати модель, де на полі рухаються хижаки та жертви. Жертви розмножуються, хижаки полюють на жертв. Програма моделює зміну популяцій.
30	Розробка консольної гри «Хрестики-нулики» з ШІ.	Створити гру «Хрестики-нулики», де гравець може грати проти комп'ютера. Реалізувати простий алгоритм для прийняття рішень штучним інтелектом.

Категорія 3: Прикладні системи та інструменти

№	Завдання	Опис завдання
31	Розробка консольного менеджера	Створити застосунок для ведення списку

	завдань (To-Do List).	справ. Програма повинна дозволяти додавати, видаляти, редагувати завдання, позначати їх як виконані та зберігати список у файл.
32	Розробка системи обліку студентів.	Написати програму для деканату, що дозволяє вести базу студентів (ПІБ, група, середній бал). Реалізувати функції додавання, пошуку, видалення та сортування.
33	Розробка консольної системи бронювання квитків.	Імітувати систему бронювання квитків (напр., у кінотеатрі). Програма має показувати схему залу, дозволяти забронювати місце та зберігати інформацію про бронювання.
34	Розробка системи «Домашня бібліотека».	Створити застосунок для каталогізації книг. Програма повинна дозволяти додавати, шукати за автором чи назвою та сортувати книги.
35	Розробка трекера особистих фінансів.	Написати програму для ведення обліку доходів та витрат. Користувач може додавати транзакції, переглядати історію та отримувати звіт за період.
36	Розробка консольної контактної книги.	Реалізувати застосунок для зберігання контактів (ім'я, номер телефону, email) з функціями додавання, пошуку, редагування та видалення.
37	Розробка планувальника розкладу.	Створити програму, куди можна вносити розклад подій на тиждень. Реалізувати функції перегляду розкладу на день/тиждень та пошуку.
38	Розробка системи обліку товарів на складі.	Написати програму, що дозволяє додавати товари, відвантажувати їх та генерувати звіт про залишки.
39	Розробка генератора паролів.	Створити утиліту, що генерує випадкові паролі заданої довжини та складності (з цифрами, різним регістром, символами).
40	Розробка кулінарної книги.	Написати програму для зберігання рецептів з можливістю додавання та пошуку за назвою або інгредієнтом.
41	Розробка програми «Щоденник».	Створити консольний щоденник, що дозволяє створювати записи з поточною датою, переглядати та шукати в них за ключовими словами.
42	Розробка системи тестування знань.	Написати програму, що завантажує з файлу тести (питання та варіанти відповідей), проводить тестування та виставляє оцінку.
43	Розробка простого калькулятора формул.	Створити калькулятор, що обчислює значення математичних або фізичних формул, запитуючи у користувача необхідні параметри.

44	Розробка системи обліку відвідуваності.	Написати програму для викладача, що дозволяє відмічати присутніх/відсутніх та переглянути статистику відвідуваності.
45	Розробка конвертера одиниць вимірювання.	Створити утиліту, що конвертує величини між різними системами (метри/милі, кг/фунти, Цельсій/Фаренгейт).
46	Розробка таймера Pomodoro.	Реалізувати консольний таймер для техніки Pomodoro: 25 хвилин роботи, 5 хвилин перерви, з можливістю налаштування інтервалів.
47	Розробка системи обліку прокату.	Написати програму для пункту прокату (напр., велосипедів). Можливості: реєстрація видачі, повернення, розрахунок вартості.
48	Розробка системи нагадувань.	Створити консольний застосунок для додавання нагадувань (текст та дата/час). При запуску програма перевіряє, чи є актуальні.
49	Розробка англо-українського словника.	Написати програму-словник, що завантажує пари слів з файлу. Користувач може шукати переклад та додавати нові слова.
50	Розробка системи інвентаризації.	Створити програму для обліку обладнання (інвентарний номер, назва) з функціями додавання, пошуку та створення звіту.

ДОДАТОК Б. Зразок титульного листа

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ФАХОВИЙ КОЛЕДЖ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ ОДЕСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ І. І. МЕЧНИКОВА»

ЦК Програмної інженерії та комп’ютерних систем та мереж

Пояснювальна записка

до курсової роботи
з освітнього компонента «Основи програмування та алгоритмічні мови»

на тему **РОЗРОБКА ПРИКЛАДНОГО КОНСОЛЬНОГО ЗАСТОСУНКУ**

Виконав: здобувач освіти ІІІ курсу, групи ……
напряму підготовки (спеціальності):
12 – Інформаційні технології
121 – Інженерія програмного забезпечення
освітньо-професійної програми:
«Інженерія програмного забезпечення»
……
Керівник Костянтин МАМУКА

Оцінка: _____

Члени комісії:

_____ _____
(підпис) (ім’я та прізвище)

_____ _____
(підпис) (ім’я та прізвище)

_____ _____
(підпис) (ім’я та прізвище)

Засвідчую, що у цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань
Здобувач освіти _____

Одеса – 2025 р.

ДОДАТОК В. Зразок завдання на курсовий проект

ВСП «ОДЕСЬКИЙ ФАХОВИЙ КОЛЕДЖ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ ОДЕСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ І. І. МЕЧНИКОВА»

Циклова комісія Програмної інженерії та комп'ютерних систем та мереж
Освітньо-кваліфікаційний ступінь фаховий молодший бакалавр
Напрям підготовки 12 Інформаційні технології
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ ЗДОБУВАЧУ ОСВІТИ

_____
(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка прикладного консольного застосунку»

керівник *Костянтин МАМУКА*
(прізвище, ім'я, по батькові)

2. Строк подання здобувачем освіти роботи 19.12.2025
3. Вихідні дані до курсової роботи:
   – ….;
   – ….;
   – ….;
   – …..
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
   - Вступ
   - Загальний розділ
     - Постановка задачі
     - Теоретичні відомості
     - Використовувані методи та технології
     - Опис алгоритмів
     - Опис програмного забезпечення
     - Результати тестування
     - Інструкцію користувача
   - Висновки
   - Перелік джерел посилання

5. Дата видачі завдання «13» 10 2025 р.

ДОДАТОК В. Зразок завдання на курсовий проект
КАЛЕНДАРНИЙ ПЛАН

№	*Назва етапів підготовки курсового проекту*	*Строк виконання етапів проекту (№ тижня)*
1.	Отримання теми курсової роботи	13.10.2025
2.	Аналіз та постановка задачі	1-2
3.	Проєктування програми	3-4
4.	Написання програми	5-7
5.	Тестування та налагодження	8
6.	Оформлення пояснювальної записки	9
7.	Здача курсової роботи на перевірку	19.12.2025
8.	Захист курсової роботи	10

Здобувач освіти _____ _____
(підпис) (ім’я та прізвище)

Керівник проекту _____ _____
(підпис) (ім’я та прізвище)