

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДУНАЙСЬКИЙ ІНСТИТУТ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
«ОДЕСЬКА МОРСЬКА АКАДЕМІЯ»
КАФЕДРА УПРАВЛІННЯ В ТРАНСПОРТНІЙ ГАЛУЗІ
ІНСТИТУТ ПРОФЕСІЙНОЇ ОСВІТИ НАПН УКРАЇНИ

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ З
ДИСЦИПЛІНИ «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Ізмаїл; Київ 2025

УДК 004.42:378.147

М 54

*Рекомендовано до друку
Кафедрою управління в транспортній галузі
Дунайського інституту Національного університету «Одеська морська академія»
(протокол № 4 від 27 жовтня 2025 р.).*

- М 54 Методичні рекомендації до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» для студентів першого (бакалаврського) рівня вищої освіти усіх форм навчання за спеціальністю 126 «Інформаційні системи та технології», освітньо-професійна програма «Інформаційні системи та технології» / укладачі М. А. Пригодій, В.І. Чимшир. – Ізмаїл; Київ: Дунайський інститут Національного університету «Одеська морська академія»; Інститут професійної освіти НАПН України, 2025. – 25 с.

УДК 004.42:378.147

<https://doi.org/10.5281/zenodo.17643702>

У методичних рекомендаціях подано мету, завдання, структуру та зміст курсової роботи, вимоги до її оформлення, порядок захисту, критерії оцінювання, а також рекомендації щодо вибору теми та організації самостійної роботи. Розроблені матеріали спрямовані на формування у студентів практичних умінь і навичок застосування принципів об'єктно-орієнтованого програмування при розв'язанні прикладних задач, розвитку алгоритмічного мислення та професійної компетентності майбутніх фахівців з інформаційних систем і технологій.

Методичні рекомендації укладено для студентів першого (бакалаврського) рівня вищої освіти усіх форм навчання за спеціальністю 126 «Інформаційні системи та технології», а також можуть бути корисними викладачам закладів вищої освіти, які забезпечують викладання дисципліни «Об'єктно-орієнтоване програмування».

Цитувати як: Пригодій, М. А., & Чимшир, В. І. (2025). *Методичні рекомендації до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» для студентів першого (бакалаврського) рівня вищої освіти усіх форм навчання за спеціальністю 126 «Інформаційні системи та технології», освітньо-професійна програма «Інформаційні системи та технології»*. ДІНУОМА; ІПО НАПН України. <https://doi.org/10.5281/zenodo.17643702>

- © М.А. Пригодій, В.І. Чимшир, 2025
- © Дунайського інституту Національного університету «Одеська морська академія», 2025
- © Інститут професійної освіти НАПН України, 2025

ЗМІСТ

ВСТУП.....	4
Розділ 1. Тематика курсових робіт	5
Розділ 2. Порядок виконання курсової роботи	6
Розділ 3. Узгодження щодо кодування програм	8
Розділ 4. Вимоги до оформлення пояснювальної записки	12
Розділ 5. Захист курсової роботи	15
ПЕРЕЛІК ПОСИЛАНЬ	16
ДОДАТКИ	17

ВСТУП

Виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» є важливим етапом підготовки студентів спеціальності 126 «Інформаційні системи та технології», оскільки дає змогу закріпити теоретичні знання та сформувати практичні навички з розробки програмного забезпечення. Курсове завдання сприяє розвитку здатності студентів застосовувати принципи об'єктно-орієнтованого програмування (ООП) для розв'язання практичних задач інформаційних систем і технологій, підвищує рівень їхньої професійної компетентності та готує до виконання складніших проєктів у майбутній професійній діяльності.

Мета курсової роботи – формування у студентів умінь і навичок проєктування, розробки та реалізації програмних модулів інформаційних систем із використанням принципів ООП, а також розвиток аналітичного та алгоритмічного мислення для ефективного вирішення інженерних і технологічних завдань.

У результаті вивчення дисципліни «Об'єктно-орієнтоване програмування» та виконання курсової роботи у студентів повинні бути сформовані наступні компетентності:

Загальні компетентності (КЗ):

КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 7. Здатність розробляти та управляти проєктами.

КЗ 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові, предметні) компетентності (КС):

КС 3. Здатність до проєктування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

КС 4. Здатність проєктувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 14. Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проєктах (стартапах).

Програмний результат навчання (ПР 3). Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проєктування і використання інформаційних систем та технологій.

Отже, курсова робота з ООП є практичним інструментом формування професійних компетентностей студентів, інтегрує теоретичні знання та практичні навички і готує майбутніх фахівців до роботи над складними інформаційними системами.

Розділ 1. Тематика курсових робіт

Тематика курсових робіт з дисципліни «Об'єктно-орієнтоване програмування» формується з урахуванням специфіки підготовки фахівців для галузі морського транспорту, логістики та суміжних сегментів морської академії. Вибір тем спрямований на поєднання теоретичних знань із практичними навичками програмування, забезпечуючи можливість студентам застосувати методи об'єктно-орієнтованого підходу для моделювання реальних систем, що використовуються у морській діяльності. При розробленні тем враховано актуальні потреби морської галузі, включаючи управління судновим та портовим персоналом, облік вантажів і контейнерів, планування рейсів, контроль безпеки та технічного стану суден, обробку даних про клієнтів і замовлення, а також автоматизацію бізнес-процесів морських компаній.

Для зручності та наочності всі теми наведені у **Додатку А**, де представлено 30 прикладів курсових робіт, у яких передбачено введення, зберігання, зміну та архівування інформації. Кожна тема описує структуру даних, операції над ними та можливі функціональні модулі програми, що дає змогу студентам оцінити складність та обсяг майбутньої роботи. Тематика також враховує різні аспекти професійної підготовки, зокрема: роботу з базами даних, формування звітів, пошук і сортування інформації, контроль за станом ресурсів та планування діяльності.

Курсові роботи повинні відповідати низці вимог, що забезпечують їх практичну цінність і відповідність навчальним результатам дисципліни. Розроблена програма має містити:

- 1) *чітко структуровану інформаційну модель*, що відповідає специфіці обраної теми;
- 2) *можливість введення, зміни та видалення даних*, включаючи контроль коректності введеної інформації;
- 3) *функціонал пошуку та відбору даних* за заданими критеріями або шаблонами;
- 4) *автоматичне формування звітів або списків*, необхідних для аналізу або прийняття рішень у контексті теми;
- 5) *архівування та облік історичних даних*, що дає змогу зберігати інформацію про виконані операції або закриті записи;
- 6) *простий та зручний інтерфейс користувача*, що забезпечує ефективну взаємодію з системою;
- 7) *застосування принципів об'єктно-орієнтованого програмування*, включаючи інкапсуляцію, наслідування та поліморфізм, для організації коду та підвищення його гнучкості.

Виконання курсової роботи відповідно до цих вимог дає змогу студентам не лише засвоїти методи ООП, а й сформувати практичні навички розроблення прикладних програм, що моделюють реальні процеси в морській галузі, сприяючи розвитку професійних компетентностей і готовності до майбутньої професійної діяльності.

Розділ 2. Порядок виконання курсової роботи

Курсова робота з дисципліни «Об'єктно-орієнтоване програмування» виконується за чітко визначеними етапами, що дають змогу студенту послідовно розвивати проєкт від ідеї до готової програми з практичним застосуванням у морській галузі. Основні етапи виконання роботи включають:

- 1) формулювання мети роботи;
- 2) визначення та опис вимог до програми;
- 3) проєктування програмного забезпечення;
- 4) кодування;
- 5) тестування;
- 6) підготовка та завершення пояснювальної записки;
- 7) захист курсової роботи.

Важливою складовою роботи є **пояснювальна записка**, що відображає всі етапи створення програми та є доказом виконання завдання у встановлені терміни. Розділи записки рекомендується розробляти паралельно з програмним кодом, або навіть попередньо, щоб зафіксувати науково-методичне обґрунтування рішень. У записці слід уникати матеріалів, що не мають прямого відношення до теми, наприклад, загальних цитат із підручників.

1. Формулювання мети роботи.

На цьому етапі студент ознайомлюється з предметною областю своєї теми, що в контексті Морської академії може включати автоматизацію обліку вантажів, планування рейсів суден, управління портовими ресурсами чи облік персоналу. У розділі «Вступ» формулюється мета курсової роботи та описується практичне застосування програми. Необхідно пояснити, як зміниться робота користувача після впровадження програмного продукту.

2. Опис вимог до програми.

На цьому етапі визначаються сценарії використання програми, її основні функції та взаємодія з користувачем. Кожен сценарій повинен містити: ідентифікатор, назву, передумови та опис основної послідовності дій користувача, а також можливі альтернативні варіанти. На основі сценаріїв формуються функції програми, що детально описуються та ілюструються ескізами користувацького інтерфейсу (англ. User Interface, **UI**). Такий опис забезпечує можливість реалізації функції будь-яким програмістом без додаткових пояснень, а також використовується для тестування.

3. Проєктування.

Етап проєктування передбачає вибір архітектури програми (консольна, настільна з GUI (Graphical User Interface) або мережева), визначення загальної структури, форматів зберігання даних та взаємодії класів. Взаємозв'язки між елементами системи відображаються за допомогою діаграм. Проєктування окремих модулів включає створення об'єктних моделей та попередній опис класів.

Завершується етап формування розділу пояснювальної записки «*Проєктування програми*».

4. Кодування програми.

На етапі кодування здійснюється реалізація вихідного коду відповідно до описаних сценаріїв та проєктної документації. Весь код включається до додатку до пояснювальної записки та повинен відповідати стандартам якості, наданим у **розділі 3** методичних рекомендацій.

5. Тестування програми.

Тестування поділяється на **модульне** та **функціональне**. *Модульне* тестування виконується під час кодування та перевіряє окремі компоненти системи. *Функціональне* тестування оцінює виконання програмою всіх заявлених функцій в умовах коректних та некоректних дій користувача. Тестування планується на етапі проєктування та проводиться по мірі реалізації функцій, завершуючись перевіркою програми у цілому.

6. Завершення пояснювальної записки.

Під час завершення роботи студент готує інструкції з розгортання та використання програми, формує висновки щодо досягнення поставленої мети та оцінює можливості подальшого вдосконалення програмного продукту. Інструкція містить перелік необхідного програмного забезпечення, покроковий опис встановлення та видалення програми. *Користувацька інструкція* детально пояснює роботу кожної функції з ілюстраціями екранних форм та алгоритмами дій. У висновках оцінюється ступінь завершеності проєкту, відповідність його заявленим цілям, а також набуті під час розробки навички.

Розділ 3. Узгодження щодо кодування програм

Етап кодування є ключовою частиною курсової роботи з дисципліни «Об'єктно-орієнтоване програмування», оскільки саме тут реалізуються теоретичні знання студентів у вигляді практичного програмного продукту. Програмна реалізація здійснюється у середовищі розробки **IntelliJ IDEA Community Edition** із використанням платформи Java на базі **Java Development Kit (JDK) 21**. Такий вибір забезпечує сумісність із сучасними вимогами до програмного забезпечення, підтримку нових синтаксичних можливостей Java, а також високий рівень зручності розроблення, налагодження та тестування.

1. Загальні положення.

Метою етапу кодування є створення робочої програми, що реалізує функціональні вимоги, визначені в попередніх розділах пояснювальної записки. Код повинен бути:

- *працездатним*, стабільним у виконанні та захищеним від типових помилок;
- *читабельним*, структурованим і зрозумілим для подальшого супроводу;
- *узгодженим* зі стандартами, прийнятими в межах освітньої програми спеціальності 126 «Інформаційні системи та технології».

Усі програмні файли мають бути розміщені у проєкті IntelliJ IDEA з дотриманням структурної ієрархії пакетів/папок (*package*). Повний вихідний код включається до додатків пояснювальної записки або подається окремо в електронному вигляді.

2. Технологічна основа курсової роботи.

У процесі виконання курсової роботи студент використовує:

JDK 21 – сучасну версію Java Platform Standard Edition, що підтримує найновіші мови конструкції (наприклад, *record*, *pattern matching*, *switch expressions*, *sealed classes* тощо).

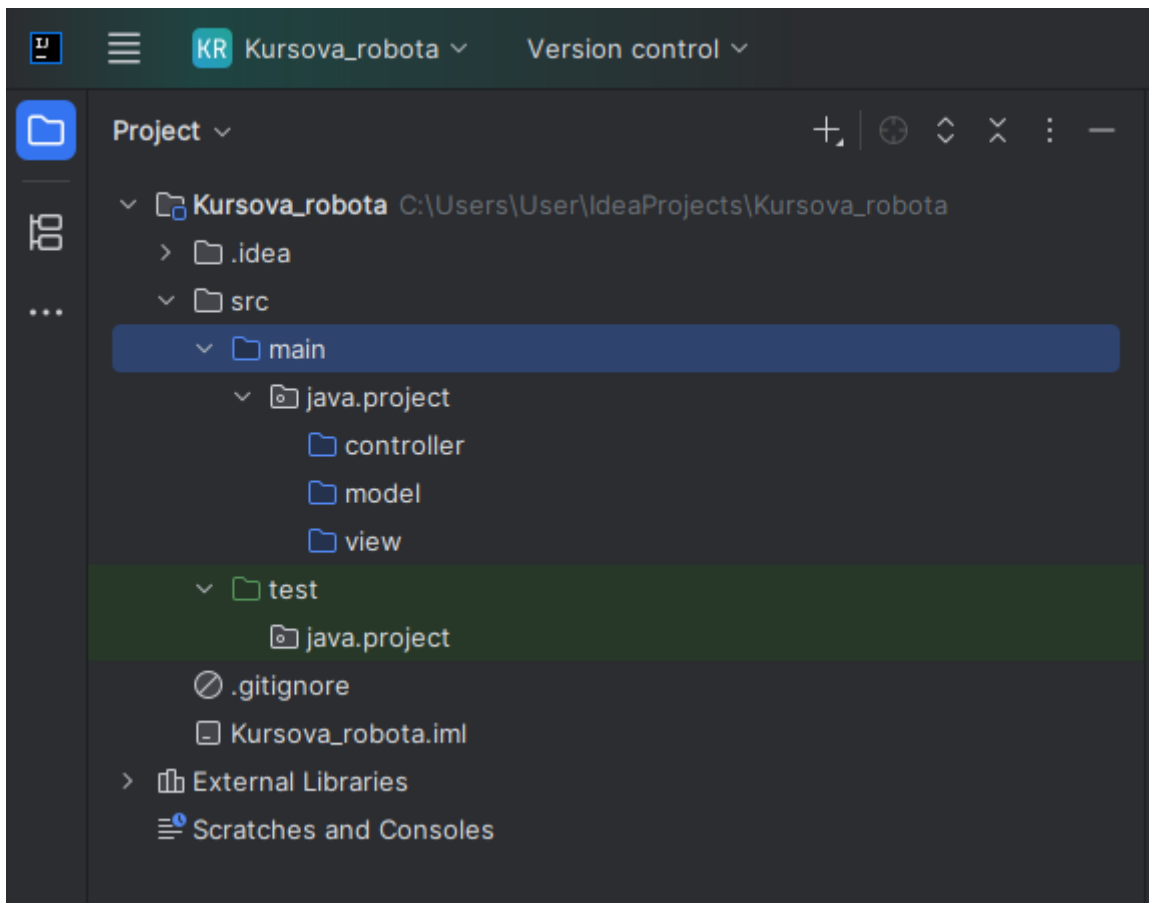
IntelliJ IDEA Community Edition – інтегроване середовище розробки, що забезпечує автоматичну компіляцію, налагодження, підсвічування синтаксису, автодоповнення коду, генерацію документації (*JavaDoc*) та роботу з системами контролю версій (*Git*).

Розроблення проводиться у відповідності до принципів об'єктно-орієнтованої парадигми, із застосуванням концепцій *інкапсуляції*, *наслідування*, *поліморфізму* та *абстракції*.

3. Вимоги до структури вихідного коду.

3.1. Структура проєкту.

Проєкт повинен містити окремі пакунки для логіки програми, графічного інтерфейсу користувача, роботи з файлами або базами даних, а також для тестів. Наприклад:



src/main/java.project/... → основний код програми

src/test/java.project/... → код тестів

3.2. Позначення:

класи – з великої літери (ShipRoute, CrewMember);

методи й змінні – у стилі camelCase (calculateDistance, portName);

постійні значення – великими літерами (MAX_SPEED);

пакети – малими літерами (java.project.navigation).

3.3. Коментарі.

Кожен клас, метод і суттєвий фрагмент коду супроводжується коментарем, який коротко пояснює його призначення. Для основних класів обов'язково генерується документація у форматі JavaDoc.

3.4. Форматування.

Відступи – 4 пробіли, рядки не перевищують 120 символів, логічні блоки відокремлюються порожніми рядками. Код має бути акуратним і візуально структурованим.

4. Вимоги до програмної реалізації.

Програма має бути створена відповідно до описаних у пояснювальній записці сценаріїв використання. Для кожного сценарію повинні бути розроблені окремі класи або модулі, що реалізують його логіку.

Не допускається:

– дублювання коду (потрібно використовувати спільні методи);

- використання застарілих або не підтримуваних API;
- ігнорування перевірки введених даних.

Особливу увагу слід приділяти роботі з виключеннями (exceptions). Усі потенційно небезпечні операції повинні бути оточені обробниками (try–catch–finally), а повідомлення користувачеві – інформативними й зрозумілими.

5. Об'єктно-орієнтований підхід.

Курсова робота повинна демонструвати застосування таких принципів ООП:

– **Інкапсуляція**. Дані класів мають бути приховані (private), доступ до них здійснюється через публічні методи (getters/setters).

– **Наслідування**. Використовується для спільних характеристик об'єктів, наприклад, клас Car може бути базовим для ElectroCar, HybridCar.

– **Поліморфізм**. Реалізується через перевизначення методів (@Override), що дає змогу гнучко адаптувати поведінку об'єктів.

– **Абстракція**. Виділення суттєвих властивостей, необхідних для роботи програми, без деталізації внутрішніх механізмів.

Демонстрація цих принципів є обов'язковою умовою для позитивного оцінювання курсової роботи.

6. Документування та супровід.

Для кожного класу обов'язково додається *JavaDoc*-коментар у форматі:

```
/**
 * Клас описує основні властивості судна та його маршрути.
 * @author Student
 * @version 1.0
 */
```

`/** ... */` – це спеціальний тип багаторядкового коментаря в Java, який називається *JavaDoc*. Він використовується для документування коду. З нього можна автоматично генерувати документацію у вигляді HTML-файлів за допомогою інструменту *javadoc*.

Документація генерується в IntelliJ IDEA (меню: *Tools* → *Generate JavaDoc*) та включається до додатку пояснювальної записки у форматі HTML або PDF.

Крім того, бажано створити *README-файл*, у якому коротко описати призначення програми, способи запуску, необхідні бібліотеки та параметри середовища виконання (JDK 21).

7. Тестування та налагодження.

Під час кодування студент проводить **модульне тестування (Unit Testing)** із використанням бібліотеки **JUnit 5**, що інтегрована до IntelliJ IDEA.

Тестові методи перевіряють:

- правильність виконання основних алгоритмів;
- коректність обробки невалідних або граничних даних;
- відповідність фактичних результатів очікуванню.

Результати тестування доцільно оформити у вигляді звіту або таблиці в додатку.

8. Якість та етичні стандарти коду.

Якість коду оцінюється за такими критеріями:

- стабільна робота програми;
- відсутність синтаксичних і логічних помилок;
- наявність коментарів і документації;
- дотримання принципів SOLID;
- відсутність дублювання.

Код має бути *оригінальним* і *створеним самостійно*. Використання зовнішніх бібліотек допускається лише у відкритих ліцензіях (наприклад, Apache 2.0, MIT) із зазначенням джерела. Забороняється копіювання чужого коду без посилань.

9. Особливості застосування в умовах Інституту чи Академії.

Ураховуючи профіль навчання, рекомендовано, щоб тематика програм відображала специфіку морської галузі – управління флотом, навігацію, логістику, навчальні симуляції, облік ресурсів чи автоматизацію розкладів.

Код має бути:

- надійним* – стабільно працювати в тривалому режимі;
- захищеним* – не допускати втрати або несанкціонованого доступу до даних;
- масштабованим* – легко адаптуватися до нових умов.

10. Оформлення вихідного коду в пояснювальній записці.

Вихідний код, який додається до пояснювальної записки, оформлюється згідно з такими вимогами:

- 1) шрифт – Courier New, 10–12 пт;
- 2) нумерація сторінок – наскрізна;
- 3) кожен файл має заголовок із зазначенням назви класу, дати створення та автора;
- 4) великі проекти можуть подаватися частково в друкованій формі, а повністю – на електронному носії або у вигляді архіву IntelliJ IDEA.

Дотримання викладених у цьому розділі вимог до кодування забезпечує високу якість програмного продукту, уніфікує підхід до розробки курсових робіт і сприяє формуванню професійних компетентностей студентів. Використання *IntelliJ IDEA Community Edition* та *JDK 21* дає змогу студентам Дунайського інституту Національного університету «Одеська морська академія» оволодіти сучасними інструментами промислової розробки, що відповідають міжнародним стандартам у сфері інформаційних технологій і транспортної логістики.

Розділ 4. Вимоги до оформлення пояснювальної записки

4.1. Структура пояснювальної записки.

Пояснювальна записка до курсової роботи є основним документом, який відображає весь процес виконання проєкту та містить наступні елементи:

- титульний аркуш (форма наведена у **додатку Б**);
- аркуш завдання та календарний план (**додаток В**);
- реферат (**додаток Г**);
- зміст;
- вступ (близько 1 сторінки);
- основні розділи: «Опис вимог», «Проектування програми», «Інструкція користувача» (загальний обсяг – 15–30 сторінок);
- висновки (близько 1 сторінки);
- перелік посилань;
- додатки (**включаючи вихідний код програми**).

На титульному аркуші зазначаються назва дисципліни, тема курсової роботи, прізвище, ім'я та по батькові студента. Реферат містить інформацію про обсяг записки (кількість рисунків, таблиць, додатків, джерел), а також короткий опис мети роботи, отриманих результатів і застосованих технологій, ключові слова.

Основні розділи пояснювальної записки повинні бути логічно структуровані на підрозділи та, за потреби, пункти. Кожен пункт повинен містити завершену думку. Обов'язковими є розділи: опис вимог, проектування програми та інструкція користувача.

4.2. Загальні вимоги.

Мова оформлення пояснювальної записки – українська або, за погодженням із керівником та завідувачем кафедри, іноземна (мова країн ЄС).

Документ оформлюється у редакторі Microsoft Word або сумісних програмах, на стандартних аркушах формату А4 (210×297 мм). Текст і таблиці друкуються чорним кольором; рисунки та схеми можуть бути чорно-білими або кольоровими.

4.3. Вимоги до тексту

Шрифт тексту: Times New Roman, пряме накреслення, розмір 14 пт.

Міжрядковий інтервал – 1,5, відступи перед та після рядка – 0 пт.

Поля сторінки: ліве – 3 см, праве – 1,5 см, верхнє і нижнє – 2 см.

Заголовки структурних розділів: **РЕФЕРАТ, ЗМІСТ, ВСТУП, основні розділи (РОЗДІЛ 1. ОПИС ВИМОГ; РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМИ; РОЗДІЛ 3. ІНСТРУКЦІЯ КОРИСТУВАЧА), ВИСНОВКИ, ПЕРЕЛІК ПОСИЛАНЬ** друкуються великими літерами, напівжирним шрифтом, без крапки в кінці. Допускається вирівнювання заголовка по центру сторінки.

Текст розділів складається з повноцінних речень. Підрозділи та пункти мають заголовки, які друкуються з абзацного відступу з великої літери без крапки в кінці.

Відстань між заголовком та текстом – два міжрядкових інтервали. Кожен розділ починається з нового аркуша. Не допускається, щоб заголовок підрозділу або пункту завершував сторінку, якщо після нього немає хоча б двох рядків тексту. Аналогічно, рисунки, таблиці або схеми не повинні завершувати підрозділ або пункт без подальшого тексту.

Використання напівжирного та курсиву у тексті обмежується: курсив рекомендується для елементів формул для однозначного трактування.

4.4. Нумерація.

Розділи, підрозділи та пункти нумеруються арабськими цифрами.

Номер підрозділу складається з номера розділу та порядкового номеру підрозділу (через крапку).

Номер пункту складається з номера розділу, номера підрозділу та порядкового номера пункту (через крапку).

Приклад:

Розділ: Розділ 1. ОПИС ВИМОГ

Підрозділ: 1.2. Назва другого підрозділу першого розділу

Пункт: 1.2.1. Назва першого пункту другого підрозділу першого розділу

Сторінки нумеруються арабськими цифрами, включаючи додатки. Номер сторінки розташовується праворуч у верхньому куті без крапки (розмір 12 пт). Титульний аркуш враховується у загальній нумерації, проте номер на ньому не проставляється.

4.5. Переліки (нумеровані та маркіровані списки).

Перед переліком ставиться двокрапка.

Переліки одного рівня позначаються тире:

- текст першого пункту;
- текст другого пункту;
- текст третього пункту.

Якщо перелік має підпорядкованість, використовують малі літери української абетки та арабські цифри, наприклад:

- | | | |
|----------------|----------------|-------------|
| а) підпункт 1; | 1) підпункт 1; | 1. Пункт 1. |
| б) підпункт 2; | 2) підпункт 2; | 2. Пункт 2. |
| в) підпункт 3. | 3) підпункт 3. | 3. Пункт 3. |

4.6 Оформлення рисунків.

Всі графічні матеріали є рисунками.

Рисунки розташовуються безпосередньо після тексту, де вони згадані, або на наступній сторінці.

Кожен рисунок має назву, розташовану під рисунком по центру, з великої літери.

Нумерація рисунків наскрізна арабськими цифрами або у межах розділу/додатку (наприклад, Рис. 3.2 або Рис. А.1).

Посилання на рисунки у тексті: «...на рисунку 3.1 наведена схема алгоритму» або «...(рис. 3.1)». При повторному згадуванні рисунку (див. рис. 3.1) Особлива увага приділяється чіткості ілюстрацій для зручності сприйняття.

4.7. Оформлення програмного коду.

Весь вихідний код проєкту наводиться у додатку до пояснювальної записки.

Невеликі фрагменти коду можуть бути включені у текст розділів для ілюстрації алгоритмів. Код повинен відповідати стандартам оформлення та бути читабельним.

Приклад:

```
public int Length {  
    get {  
        if (next == null)  
            return 1;  
        else  
            return next.Length + 1;  
    }  
}
```

4.8. Складання переліку джерел.

Перелік джерел складається відповідно до ДСТУ 8302:2015. Бібліографічні описи наводяться в порядку першого згадування в тексті. Посилання на джерела у тексті подаються в квадратних дужках, наприклад:

[5] – п'яте джерело;

[5, с. 6] – шоста сторінка п'ятого джерела;

[5, с. 6–8] – інформація розміщується на шостій, сьомій та восьмій сторінках п'ятого джерела;

[5; 6] – п'яте та шосте джерело;

[5; 6] – п'яте та шосте джерело;

[5, с. 7; 6, с. 12] – сьома сторінка п'ятого джерела та дванадцята сторінка шостого джерела.

4.9. Додатки

Додатки оформлюються як продовження записки, на окремих сторінках.

Кожен додаток має заголовок із великої літери, симетрично до тексту.

На середині рядка вгорі сторінки над заголовком зазначають слово «Додаток» та літеру, що його позначає (А, Б, В ...). Букви Ґ, Є, З, І, Ї, Й, О, Ч, Ъ не використовуються.

Обов'язковим додатком є **вихідний код програми**.

Розділ 5. Захист курсової роботи

Допуск студента до захисту курсової роботи здійснюється викладачем, який керує курсовою роботою, і відбувається за наявності виконаних обов'язкових умов. До основних умов допуску належать:

- 1) наявність діючої програми, що повністю відповідає проєктному завданню;
- 2) правильно оформлений і структурований вихідний код програми;
- 3) відповідність пояснювальної записки вимогам, викладеним у розділі 4 та інших частинах методичних рекомендацій;
- 4) створення необхідних каталогів у Google Клас «Курсова робота ООП», завантаження всіх обов'язкових файлів із правильними іменами, проходження перевірки на плагіат.

Захист курсової роботи відбувається перед комісією, до складу якої входять три викладачі, у присутності студентів академічної групи автора.

Під час виступу студент повинен:

1. Коротко представити тему та мету курсової роботи.
2. Викласти основні результати та функціональні можливості створеної програми (з представленням User Interface).
3. Продемонструвати роботу програмного продукту та пояснити алгоритми реалізації ключових функцій.

Орієнтовний час виступу – **10 хвилин**. Після доповіді студент відповідає на запитання членів комісії, які стосуються як технічної частини роботи, так і оформлення пояснювальної записки та використаних методів.

Якщо під час захисту або перевірки буде виявлено, що робота виконана не самостійно, студент отримує оцінку «незадовільно».

Курсова робота оцінюється за наступними критеріями:

Якість програмного продукту та користувацького інтерфейсу – оцінюється функціональність програми, її стабільність, наявність усіх заявлених функцій та зручність користування.

Якість вихідного коду – оцінюється структурованість, коментування коду, дотримання стандартів кодування та принципів ООП.

Виступ та відповіді на запитання – оцінюється логічність доповіді, вміння аргументовано відповідати на питання, демонструвати практичну користь розробки.

Якість пояснювальної записки – оцінюється відповідність вимогам оформлення, повнота розділів, зрозумілість опису сценаріїв використання та проєктних рішень.

Відповідність термінів виконання етапам календарного плану – оцінюється своєчасність виконання всіх етапів проєкту, включаючи розробку, тестування та підготовку документації.

Захист курсової роботи є підсумковим етапом оцінювання, що дає змогу перевірити як практичні навички студента, так і його теоретичну підготовку.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання». URL: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf
2. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання»/ URL: https://kubg.edu.ua/images/stories/podii/2017/06_21_posylannia/dstu_8302.pdf
3. Об'єктно-орієнтоване програмування : навчал. посібник / А. Л. Рачинська, О. А. Недева, К. С. Палій. Одеса : Одеський нац. ун-т ім. І. І. Мечникова, 2024. 338 с. URL: <https://dspace.onu.edu.ua/handle/123456789/38083>
4. Об'єктно-орієнтоване програмування : підручник / В. В. Бублик. Київ : ІТ-книга, 2015. 624 с. URL: https://itknyga.com.ua/documents/OOP_final.pdf
5. Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. Київ : КНУ ім. Ігоря Сікорського, 2019. 192 с. URL: https://ela.kpi.ua/bitstream/123456789/53747/1/OOP_Alkhimova_KR_pidruchnyk.pdf
6. Основи об'єктно-орієнтованого програмування : навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. Харків : ХНЕУ ім. С. Кузнеця, 2019. 237 с. URL: <https://repository.hneu.edu.ua/bitstream/123456789/23847/1/2019-%D0%A9%D0%B5%D1%80%D0%B1%D0%B0%D0%BA%D0%BE%D0%B2%20%D0%9E%20%D0%92.pdf>
7. Програмування – 2. Об'єктно-орієнтоване програмування : навч. посібник / В. Б. Бобков, Ю. Є. Грудзинський, К. В. Крилов. Київ : НТУУ «КПІ», 2023. 77 с. URL: <https://ela.kpi.ua/handle/123456789/57281>

ДОДАТКИ

Додаток А

Приблизний перелік тем курсових робіт

1. Картотека суднових екіпажів.

Дані по кожному члену екіпажу: прізвище, ім'я, посада на борту, кваліфікація, громадянство, останній рейс, термін контракту, медичні сертифікати, наявність спеціальної підготовки та навичок. Система повинна забезпечувати введення нових членів екіпажу, зміну даних (наприклад, підвищення кваліфікації або зміна посади), пошук за різними параметрами (посада, досвід, сертифікати), формування списків для конкретного рейсу, а також архівування випущених зі служби або звільнених членів екіпажу.

2. Система обліку сертифікацій морських фахівців.

Дані про курси, іспити, сертифікати та дати їх дії для офіцерів і членів екіпажу. Реалізація функцій перевірки термінів дії сертифікатів, автоматичне нагадування про необхідність повторного навчання або перепідготовки, вибірка сертифікатів за типом або посадою, формування статистичних звітів за професійним рівнем, архівування старих або прострочених сертифікатів.

3. База кандидатів на роботу у морських компаніях.

Дані про кандидатів: персональна інформація, досвід роботи, освіта, знання мов, готовність до рейсів, рекомендації. Можливість фільтрувати кандидатів за критеріями, формувати списки для співбесід та відбору на конкретні посади, архівування відмовлених кандидатур або тих, хто пройшов відбір. Реалізація модулів статистики по ринку праці та заповнюваності вакансій.

4. Система оцінки ефективності роботи екіпажу.

Статистика рейсів, виконання поставлених завдань, професійні досягнення, порушення чи інциденти. Введення та оновлення даних після кожного рейсу, пошук за членом екіпажу, формування рейтингу ефективності та звітів для керівництва компанії. Архівування завершених періодів оцінки, що дає змогу аналізувати динаміку ефективності роботи.

5. Журнал медичних оглядів екіпажу.

Дані медичних обстежень: дати проведення, результати, рекомендації, історія змін стану здоров'я, алергії та медичні обмеження. Реалізація пошуку за членом екіпажу та типом обстеження, нагадування про повторні огляди, архівування старих даних та формування звітів для медичної служби.

6. Облік суден компанії.

Дані суден: назва, тип, тоннаж, технічний стан, розклад рейсів, екіпаж, статус доступності. Система повинна дозволяти додавати нові судна, змінювати інформацію про технічний стан чи розклад, шукати судна за типом або маршрутом, архівувати списані чи продані судна. Можливе формування статистики завантаженості флоту.

7. Система управління технічним обслуговуванням суден.

Графіки планового ТО, історія виконаних робіт, наявність запасних частин та матеріалів, відповідальні особи. Введення та оновлення даних, формування нагадувань про обслуговування, пошук за судном або датою, архівування завершених робіт та створення аналітичних звітів про стан флоту.

8. База аварій та інцидентів на суднах.

Опис події, дата, екіпаж, причини, наслідки, залучені ресурси. Система дозволяє вводити нові інциденти, змінювати статус розслідування, проводити пошук за судном або типом інциденту, формувати звіти для керівництва та архівувати закриті випадки.

9. Реєстр суден у відкритому морі.

Дані про розташування суден: координати, швидкість, курс, тип вантажу, стан обладнання, інциденти в рейсі. Введення даних у реальному часі, пошук та відображення маршрутів, формування історії переміщень та архівування завершених рейсів для аналізу логістики та безпеки.

10. Система оцінки економічної ефективності судноплавства.

Вантажооборот, витрати палива, прибутковість рейсів, доходи та витрати компанії. Система дозволяє вводити дані після рейсу, змінювати показники при уточненні, проводити аналіз за періодами, формувати звіти для керівництва та архівувати завершені фінансові періоди.

11. Система управління вантажними перевезеннями.

Дані про вантажі: найменування, вага, обсяг, відправник, отримувач, судно, маршрут, статус доставки, дати завантаження та прибуття. Система повинна забезпечувати введення та зміну даних про вантажі, пошук за критеріями (тип, відправник, маршрут), формування накладних і звітів, архівування доставлених вантажів.

12. База контейнерних перевезень.

Інформація про кожен контейнер: номер, тип вантажу, стан контейнера, термін доставки, розташування на судні або в порту. Можливість введення нових контейнерів, зміни статусу та переміщення, пошук контейнерів за різними параметрами, формування плану відправок, архівування доставлених контейнерів.

13. Клієнтська база морського агентства.

Дані замовників: контактна інформація, історія замовлень, поточний статус перевезень, фінансові умови, пріоритети. Система дозволяє вводити нових клієнтів, змінювати умови контрактів, шукати за різними критеріями, формувати звіти та архівувати завершені замовлення.

14. Система обліку пасажирів круїзних суден.

Інформація про пасажирів: персональні дані, бронювання, поселення, маршрути, харчування та додаткові послуги. Введення нових бронювань, редагування інформації, пошук пасажирів за різними параметрами, формування списків для посадки та архівування завершених круїзів.

15. Моніторинг та планування рейсів пасажирських суден.

Розклад, проміжні порти, наявність вільних місць, тривалість рейсу. Можливість введення та зміни розкладу, пошук оптимальних рейсів, бронювання та формування посадкових документів, архівування завершених рейсів для статистики.

16. База даних портових терміналів.

Дані про порти: розташування, довжина причалу, вантажопідйомність, доступність для різних типів суден, наявність послуг. Система дозволяє додавати нові порти, змінювати параметри, пошук портів за характеристиками, формування планів розвантаження та архівування застарілих даних.

17. Система управління контейнерним терміналом.

Дані контейнерів: власник, вантаж, розмір, термін зберігання, місце на складі. Система забезпечує введення нових контейнерів, зміну розташування, пошук за власником або типом вантажу, формування накладних, архівування відправлених контейнерів.

18. Облік вантажообігу в порту.

Дані про прибуття і відправку вантажів: вага, обсяг, перевізник, тип вантажу, терміни доставки. Введення та зміна інформації, пошук вантажів за різними параметрами, формування звітів та архівування завершених операцій.

19. Контроль доступу суден до порту.

Реєстрація суден, дозвіл на швартування, облік вільних та зайнятих причалів, пріоритетність доступу. Система дозволяє вводити нові заявки, змінювати статус дозволів, шукати за судном або датою, архівувати оброблені запити.

20. Система планування черг на портові послуги.

Вибір суден для обслуговування, пріоритетність обслуговування, контроль черги, формування графіків. Введення нових даних, зміна пріоритетів, пошук та відображення поточної черги, архівування виконаних операцій для аналізу ефективності роботи порту.

21. База даних замовників морських перевезень.

Дані про замовників: контактні дані, історія замовлень, фінансові умови, статус виконання перевезень. Введення та редагування даних, пошук за замовником або типом перевезення, формування звітів та архівування завершених контрактів.

22. Система підбору оптимального маршруту вантажу.

Дані про маршрути: порти відправлення і прибуття, відстань, час, умови безпеки, наявність ресурсів. Система забезпечує введення нових маршрутів, пошук оптимальних варіантів за критеріями, формування планів перевезень, архівування використаних маршрутів для аналітики.

23. Облік тарифів та комісій морських агентств.

Дані про вартість перевезень, знижки, комісійні, рахунки, умови оплати. Введення та зміна тарифів, пошук за клієнтом або рейсом, формування фінансових звітів, архівування завершених операцій.

24. Управління договірною документацією.

Реєстрація договорів, терміни дії, автоматичні нагадування, зміни умов. Система дозволяє додавати нові договори, редагувати існуючі, шукати за номером чи замовником, архівувати закриті або завершені договори.

25. База страхових випадків при перевезеннях.

Опис інциденту, судно, вантаж, сума компенсації, статус розслідування. Введення нових випадків, зміна статусу, пошук за судном або вантажем, формування звітів для страхових компаній, архівування закритих випадків.

26. Система моніторингу погодних умов для морських рейсів.

Дані про погоду: температура, вітер, хвилі, штормові попередження. Система забезпечує введення прогнозів, пошук даних за датою та координатами, формування рекомендацій для планування рейсів, архівування старих прогнозів.

27. База портових послуг для судновласників.

Дані про послуги: буксирування, завантаження/розвантаження, технічна допомога, ціни, умови надання. Введення нових послуг, зміна параметрів, пошук за типом послуги, формування заявок та архівування виконаних.

28. Система контролю безпеки на борту.

Інциденти, доступ до інформації, система звітності, статистика. Введення нових випадків, зміна статусу, пошук за судном або типом інциденту, формування звітів, архівування закритих випадків.

29. Облік суднових запасів і палива.

Наявність запасів: паливо, запчастини, витратні матеріали, термін придатності. Введення нових поставок, зміна кількості, формування плану поповнення, пошук за ресурсом, архівування використаних запасів.

30. Система планування практики студентів морської академії.

Розклад практики, контроль відвідуваності, призначення наставників, архів минулих практик. Введення нових студентів та рейсів, зміна розкладу, пошук за студентом або датою, формування звітів та архівування завершених практик.

Додаток Б
Зразок титульного аркушу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДУНАЙСЬКИЙ ІНСТИТУТ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
«ОДЕСЬКА МОРСЬКА АКАДЕМІЯ»
КАФЕДРА УПРАВЛІННЯ В ТРАНСПОРТНІЙ ГАЛУЗІ

КУРСОВА РОБОТА
з дисципліни «Об'єктно-орієнтоване програмування»

Тема: «[Повна назва теми курсової роботи]»

Виконавець:

студент [курс, група]
[Прізвище Ім'я По батькові]

Керівник роботи:

[Науковий ступінь, звання,
Прізвище Ім'я По батькові]

Ізмаїл 2025

Додаток В
Аркуш завдання та календарний план

ДУНАЙСЬКИЙ ІНСТИТУТ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
«ОДЕСЬКА МОРСЬКА АКАДЕМІЯ»
КАФЕДРА УПРАВЛІННЯ В ТРАНСПОРТНІЙ ГАЛУЗІ

Рівень вищої освіти: *перший (бакалаврський)*

Дисципліна: **Об'єктно-орієнтоване програмування**

Спеціальність: **126 «Інформаційні системи та технології»**

Освітньо-професійна програма: **Інформаційні системи та технології**

Курс: _____. Семестр: _____. Група: _____.

ЗАВДАННЯ
до курсової роботи студента

(Прізвище, Ім'я, По батькові)

1. Тема роботи/проєкту: _____

2. Термін здачі закінченого проєкту: «____» _____ 202__ р.

3. Вихідні дані до проєкту (специфікація програми): _____

(опис того, що програма повинна робити: її функції, обмеження, вимоги до користувача та системи)

4. Зміст розрахунково-пояснювальної записки:

Реферат – короткий виклад змісту роботи, включно з метою проєкту, основними результатами, застосованими технологіями та ключовими словами.

Зміст – структурований перелік розділів, підрозділів та додатків із зазначенням сторінок.

Вступ – опис актуальності теми, мети та завдань курсового проєкту, а також очікуваного ефекту від використання програми.

Розділ 1. Опис вимог – формалізація функціональних та нефункціональних вимог до програми, опис сценаріїв використання та функцій.

Розділ 2. Проєктування програми – опис архітектури, структури, взаємодії класів та елементів системи, діаграми та ескізи інтерфейсу користувача.

Розділ 3. Інструкція користувача – покрокове керівництво для роботи з програмою, опис функцій та приклади використання.

Висновки – підсумки виконаної роботи, оцінка досягнення поставлених цілей, пропозиції щодо вдосконалення та подальшого розвитку програмного продукту.

Перелік посилань – оформлений згідно з ДСТУ 8302:2015, включає всі джерела, на які є посилання в тексті записки.

Додатки – додаткові матеріали, наприклад, вихідний код, діаграми, графіки, таблиці тощо.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапу	Термін виконання
1	Видача теми, узгодження і затвердження теми	
2	Формулювання вимог до програми	
3	Розробка підсистеми зберігання та пошуку даних	
4	Розробка функцій програми	
5	Розробка функцій зберігання та завантаження даних	
6	Тестування і доопрацювання програмної системи	
7	Оформлення пояснювальної записки, додатків та графічного матеріалу	
8	Захист курсової роботи	

Студент _____
(підпис)

« ____ » _____ 202 ____ р.

Керівник _____
(підпис)

« ____ » _____ 202 ____ р.

Додаток Г
Зразок оформлення реферату

РЕФЕРАТ

Курсова робота присвячена розробленню програмного забезпечення для автоматизації управління річковими судновими маршрутами. Обсяг розрахунково-пояснювальної записки складає 28 сторінок, містить 6 рисунків, 4 таблиці, 2 додатки та 15 джерел посилань.

Метою роботи є створення програми, що дає змогу ефективно управляти даними про судна та їх маршрути, забезпечує зручний інтерфейс для введення, редагування та збереження інформації, а також реалізує пошук і фільтрацію даних за різними критеріями.

Програма створена з використанням об'єктно-орієнтованого підходу, розроблена на мові Java у середовищі IntelliJ IDEA Community Edition з застосуванням JDK 21.

В результаті отримана програма, що дає змогу:

- вводити та редагувати інформацію про судна та маршрути;
- здійснювати пошук і сортування даних за заданими параметрами;
- переглядати результати у зручному табличному форматі;
- експортувати та зберігати дані для подальшого використання;
- забезпечувати коректність введених даних і контроль за допустимими значеннями.

Застосовані технології та принципи об'єктно-орієнтованого програмування дають змогу підвищити надійність та гнучкість програмного продукту. Отримані результати свідчать про відповідність розробленої програми поставленим цілям та завданням курсової роботи.

Ключові слова: програмування, об'єктно-орієнтоване програмування, Java, IntelliJ IDEA, JDK 21, суднові маршрути, управління даними, програмний продукт.

Редактор-коректор – М. А. Пригодій
Бібліографічний редактор – В. О. Маркова
Формат 210/297.
Авт. арк. 1,0. Зам. 14/2025.

Виготівник і видавець:
Інститут професійної освіти НАПН України
03045 Київ, пров. Віто-Литовський, 98-а
+380 (44) 252-71-75; +380 (44) 259-45-53(факс)
e-mail: info@ivet.edu.ua

Підтвердження внесення суб'єкта видавничої справи до державного реєстру видавців,
виготівників і розповсюджувачів видавничої продукції.
Серія ДК, No 3805 від 21.06.2010 р.

Інститут професійної освіти НАПН України є суб'єктом у сфері онлайн-медіа
(Рішення Національної ради з питань телебачення і радіомовлення № 1430 від 16.11.2023 р.)