

Міністерство освіти та науки України
Національний технічний університет України “КПІ”
Кафедра ІІІ

МЕТОДИЧНІ ВКАЗІВКИ

до курсової роботи

з дисципліни

«Основи програмування 2. Модульне програмування»

Київ 2022

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	5
2.1 Об'єктно-орієнтований аналіз предметної області	5
2.2 Об'єктно-орієнтоване проектування	5
2.3 Об'єктно-орієнтоване програмування	6
2.4 Тестування програми	7
2.5 Оформлення пояснювальної записки.....	7
2.6 Захист курсової роботи.....	8
3 ЗМІСТ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	9
4 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	14
4.1 Загальні вимоги	14
4.2 Нумерація.....	15
4.3 Оформлення цитат і переліку посилань.....	16
4.4 Оформлення додатків	17
5 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОЇ РОБОТИ.....	18
5.1 Хід виконання та захисту курсової роботи	18
5.2 Критерії оцінювання курсової роботи	19
6 ТЕМИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ НА КУРСОВУ РОБОТУ	24
6.1 Варіанти завдань	24
6.1.1 Тематика базових індивідуальних завдань	24
6.1.2 Тематика додаткових індивідуальних завдань	31
СПИСОК ЛІТЕРАТУРИ.....	43
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ	44
ДОДАТОК Б ТИТУЛЬНА СТОРІНКА	46
ДОДАТОК В ЛИСТ ЗАВДАННЯ.....	47
ДОДАТОК Г ТЕКСТИ ПРОГРАМНОГО КОДУ.....	49

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Виконання курсової роботи (КР) з дисципліни «Основи програмування -2. Основи об'єктно-орієнтованого програмування» є обов'язковою складовою навчального плану підготовки бакалаврів за спеціальністю 121 «Інженерія програмного забезпечення».

Головна мета виконання КР полягає у закріпленні, поглибленні та узагальненні базових теоретичних знань, якими студент оволодів під час вивчення дисципліни «Об'єктно-орієнтоване програмування», їх застосуванні до комплексного вирішення конкретного фахового завдання. Такий підхід повністю відповідає концепції формування висококваліфікованих фахівців у галузі природничих наук, котрі набувають не тільки знань, але й навичок та вмінь, якими повинні володіти випускники вищого навчального закладу за спеціальністю 121 «Інженерія програмного забезпечення».

Основними цілями написання курсової роботи є:

- отримання практичного досвіду об'єктно-орієнтованого аналізу і проектування складних програмних додатків;
- набуття практичних навичок програмування на мовах високого рівня з використанням методології об'єктно-орієнтованого програмування;
- отримання навичок реалізації графічного інтерфейсу користувача;
- закріплення навичок самостійної роботи.

Робота виконується студентами усіх форм навчання протягом третього семестру.

Під час виконання курсової роботи студент 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;
- здатність доводити розв'язання поставленої задачі до логічного завершення;
- вміння аналізувати отримані результати і робити відповідні висновки.

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

2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основні етапи виконання курсової роботи:

- об'єктно-орієнтований аналіз предметної області задачі;
- об'єктно-орієнтоване проектування архітектури програмної системи;
- об'єктно-орієнтоване програмування;
- тестування програми;
- оформлення пояснювальної записки;
- захист курсової роботи.

2.1 Об'єктно-орієнтований аналіз предметної області

На цьому етапі проводиться *бізнес-аналіз* предметної області задачі. В процесі його проведення необхідно виявити:

а) Чому присвячена предметна область, які в ній є терміни і поняття, суб'єкти та об'єкти, способи взаємодії суб'єктів, способи використання об'єктів, закономірності.

б) Що входить в словник предметної області, окремо виділивши список іменників і список дієслів, які можуть бути пов'язані з іменниками.

в) Які функціональні вимоги до розроблюваної системи.

Результат аналізу повинен бути формалізований у вигляді реферативного опису предметної області. З цього опису повинен логічно слідувати словник предметної області, що складається зі списку іменників і дієслів. Саме він послужить основою наступного етапу роботи.

Вся ця інформація узагальнюється у вигляді діаграми прецедентів.

Також розробляється сценарій поведінки об'єктів - послідовний опис дій під час функціонування системи.

2.2 Об'єктно-орієнтоване проектування

На даному етапі розробляється архітектура системи.

Даний етап починається з обробки словника предметної області. Ця обробка полягає у виявленні того, які слова відповідають об'єктам, класам, властивостям і

методам. Список іменників служить основою для виділення класів і їх властивостей, а список дієслів - для визначення методів.

Наступний крок полягає у визначенні характеристик (атрибутів і методів) виділених класів. Слід звернути увагу на те, що набори атрибутів і методів у різних класів можуть «перетинатися».

Ще одним питанням, що вимагає вирішення на даному етапі, є виявлення відносин між класами. Мова йде про відносини успадкування та включення. Слід звернути увагу на те, що поняття «успадкування» найчастіше виникає тоді, коли різні класи мають частково схожі набори властивостей і методів.

При складанні словника предметної області далеко не завжди в список можуть потрапити поняття, яким можна зіставити базові класи в ієрархії успадкування. Тому слід уважно проаналізувати список класів, властивостей, методів, їх відповідність, і, можливо, виділити ряд нових класів, пов'язаних з наявними відносинами спадкування і включення.

На даному етапі можна вже враховувати не тільки особливості предметної області, а й такі принципи об'єктно-орієнтованого підходу як абстракція, інкапсуляція, поліморфізм.

Результати проектування повинні бути оформлені у вигляді *діаграми класів*. Для цього можна використати нотацію мови UML.

2.3 Об'єктно-орієнтоване програмування

На цьому етапі відбувається програмна реалізація класів.

Спроектowana діаграма класів дозволяє дуже легко написати програмний код, що містить загальний опис класів (ієрархію класів, властивості, прототипи методів). Тому основна увага в ході програмної реалізації системи класів приділяється алгоритмізації і програмуванню методів класів. Інтерфейс і реалізація кожного класу мають міститися у окремих файлах.

Також розробляється тестовий додаток, що запускає процес моделювання та перевіряє правильність створених інтерфейсів класів. Даний додаток повинен мати віконний інтерфейс, що включає меню і інтерактивні інтерфейсні засоби, за допомо-

гою яких можна реалізувати визначену функціональність додатку. Лістинги повинні бути детальним чином документовані.

2.4 Тестування програми

На цьому етапі проводиться тестування розробленого програмного забезпечення.

Слід розрізняти процеси тестування і налагодження програмного коду, яке виконується програмістом за допомогою вбудованих засобів середовища розробки і зводиться, в основному, до виявлення синтаксичних і семантичних помилок в тексті програми.

Тестування - це процес, що вимагає планування і виконання ряду попередніх процедур, основною з яких є складання набору тестових прикладів, що утворюють тест-план. Тестові приклади в більшості випадків ґрунтуються на функціональних вимогах до системи і можуть зачіпати різні рівні розробки (модульне тестування, інтеграційне тестування, системне тестування).

Методика тестування розробленого програмного забезпечення передбачає виконання спрощеного варіанта модульного тестування, що зводиться до тестування всіх методів розроблених класів. Тому план тестування повинен містити список різних варіантів виклику методів класів з очікуваними результатами і порядок тестового виконання цих викликів. Оскільки деякі методи можуть виявитися однотипними, то тести для їх перевірки можна згрупувати, щоб на наступному етапі врахувати це при проектуванні тестового додатка.

Після успішного тестування програми можна переходити до наступних етапів. В разі невдалого тестування треба повернутись до попередніх етапів курсового проектування. Результати тестування та виявлені обмеження програми необхідно задокументувати.

2.5 Оформлення пояснювальної записки

Пояснювальна записка оформлюється згідно вимог викладених у третьому та четвертому розділах даних методичних вказівок.

2.6 Захист курсової роботи

Захист курсової роботи передбачає демонстрацію студентом повністю завершеної програми та оформленої пояснювальної записки. Крім того студент має показати володіння матеріалом курсової роботи.

3 ЗМІСТ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Основними документами, що представляють КР, є пояснювальна записка та комплекс програм на CD-диску. Текст пояснювальної записки до курсової роботи повинен бути викладений лаконічно, у обґрунтованому стилі. Не дозволяється переписування літературних джерел та використання не опрацьованих студентом Інтернет-оглядів.

Пояснювальна записка виконується на аркушах формату А4 згідно ДСТУ 3008-95. У випадку необхідності окремі ілюстрації можуть виконуватись на аркушах більших форматів.

Обов'язковими структурними частинами пояснювальної записки є:

- титульний лист;
- лист-завдання на курсову роботу;
- анотація;
- зміст;
- вступ;
- основна частина;
- висновок;
- перелік посилань;
- додатки до пояснювальної записки.

Титульний лист повинен бути встановленого зразку. На ньому вказується назва міністерства, університету, інституту, факультету, кафедри і тема курсової роботи (у точній відповідності із індивідуальним завданням). Його зразок наведений у *Додатку Б*. Титульний лист не нумерується як розділ, не вноситься до змісту і не нумерується як сторінка.

Лист завдання на курсову роботу видається і заповнюється керівником. Заповнення листа завдання здійснюється викладачем при кожній консультації та здачі КР. Бланк завдання на курсову роботу повинен бути оформлений за формою, наведеною в *Додатку В*. На ньому обов'язково повинна бути вказана дата видачі завдан-

ня. Індивідуальне завдання засвідчується підписом керівника КР. Лист завдання не нумерується як розділ.

В **анотації** у реферативному стилі наводиться інформація про зміст та результати, що отримані в курсовій роботі. Як розділ анотація не нумерується.

Зміст характеризує структуру КР. Він повинен вміщувати в собі назви усіх розділів, підрозділів, пунктів та підпунктів курсової роботи, а також перелік додатків. Усі назви повинні бути записані так само як вони сформульовані в КР. Визначення сторінок у змісті обов'язкове. Зміст розміщується на окремій сторінці, як розділ зміст не нумерується. Посилання на зміст, у сам зміст не включається.

У **вступі** коротко розкривається призначення КР, ціль роботи, сутність вирішуваної задачі. Вступ як розділ не нумерується.

Основна частина пояснювальної записки може складатися з розділів, підрозділів, пунктів, підпунктів*. Кожний розділ починають з нової сторінки. У розділах основної частини подають:

- постановку задачі;
- аналіз предметної області;
- опис архітектури програмної системи;
- опис програмного забезпечення;
- результати його тестування;
- інструкцію користувача.

У **постановці задачі** висвітлюється інформаційна сутність задачі – вся інформація, необхідна для програмної реалізації технічного завдання (*Додаток А*).

Основною метою даного розділу є аналіз специфікації вимог технічного завдання на КР і формулювання додаткових вимог, які безпосередньо впливають з нього та мети КР.

Результатом аналізу є конкретизація постановки завдання, структури вхідних і вихідних даних, основних операцій і методів, які для них використовуються, основних параметрів та функціональних можливостей програмної системи, що в контексті

*Дотримання в курсовій роботі запропонованої рубрикації не є обов'язковою вимогою.

основних вимог визначає загальну концепцію розроблення, основні засоби реалізації, інтерфейс із користувачем, робоче середовище.

У **розділі аналізу** необхідно представити реферативний опис предметної області. Дана інформація узагальнюється у вигляді діаграми прецедентів системи. Також повинен бути розроблений сценарій роботи програми.

Розділ проектування архітектури програмної системи повинен містити проектну специфікацію програми у вигляді *діаграми класів* та її опису. Для представлення діаграми класів краще використати UML-нотацію.

Опис програмного забезпечення включає перелік класів та специфікацію їх методів. Даний опис здійснюється у вигляді, наведеному у таблиці 3.1.

Таблиця 3.1 – Опис класів та їх методів

№ п/п	Назва класу	Назва методу	Призначення методу	Опис вхідних параметрів	Опис вихідних параметрів	Заголовний файл

Необхідно описати усі класи які використовуються в програмі, у тому числі і стандартні.

Повний вихідний програмний код з визначеннями розроблених методів класів повинен бути поміщений у додаток (додаток Г). Код тестової програми також наводиться у додатку (додаток Г).

У розділі **тестування** треба описати методику тестування програмної системи; представити план тестування, який включає набори відповідних тестових даних та очікувані результати; навести тести, виконання яких дозволяє пересвідчитись у правильності 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. Якщо для відображення роботи програми необхідна значна кількість роздруківок, то їх можна подати в додатках.

У **інструкції користувача** описується призначення програми; вимоги до системи; наводиться докладна інструкція по роботі з програмою, в якій описується склад програмного забезпечення (імена всіх файлів, з яких складається програма із зазначенням їх розміру і призначення), варіанти використання програми – опис інтерфейсу (зовнішній вигляд, засоби керування та їх призначення) та послідовність дій для виконання тієї чи іншої функції програми.

Описуючи інтерфейс користувача, обов'язково використовувати рисунки.

У **висновках** в реферативній формі повинні бути описані результати, отримані студентом на кожному із етапів виконання роботи, а також висновки щодо досягнення мети курсової роботи. Тут необхідно наголосити на якісних і кількісних показниках здобутих результатів, обґрунтувати їх достовірність. Висновки як розділ не нумеруються.

Перелік посилань повинен включати усі літературні джерела, на які є посилання у тексті пояснювальної записки. Список повинен формуватися в порядку посилань за текстом і вміщувати бібліографічні відомості офіційно виданих книжок, статей, патентів, депонованих рукописів тощо. Як розділ перелік літератури не нумерується.

В **додатки** включають технічне завдання, лістинги бібліотеки класів та моделюючої програми, допоміжний матеріал: таблиці, графіки, скріншоти тощо, які не увійшли до пояснювальної записки, але потрібні для пояснень.

Технічне завдання (*Додаток А*) повинне бути першим з додатків. Воно формулюється у повному вигляді; перелік необхідних його пунктів визначається розробником.

У *додатку Г* повинні міститися лістинги заголовних файлів і моделюючої програми.

Текст програми (класів, методів) повинен мати коментарі. Наводиться призначення усіх ідентифікаторів (імена констант, змінних, типів даних), які використовую-

ються у програмі. Кожен метод повинен бути документованим із зазначенням не тільки його призначення, а й опису аргументів (параметрів). При підключенні власних заголовних файлів необхідно вказати в коментарях засоби відповідної бібліотеки, які будуть використовуватися у програмі. На початку кожного заголовного файлу користувача необхідно вказати в коментарях його призначення.

У інші додатки за необхідності виносять таблиці, графіки, скріншоти тощо, які не увійшли до пояснювальної записки.

Як розділ додатки не нумеруються, але кожен з додатків нумерується великими літерами алфавіту згідно ДСТУ 3008-2015.

4 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

4.1 Загальні вимоги

Пояснювальна записка має бути представлена в електронному та друкованому вигляді.

Електронна версія зберігається в банку даних кафедри. Файл із копією курсової роботи здається на кафедру разом із друкованим примірником безпосередньо під час захисту. Формат файлу – **docx**.

Текстову частину роботи необхідно друкувати на одному боці аркуша білого паперу формату А4 (210×297 мм). В окремих випадках, для більш наочного подання таблиць та ілюстрацій, можна використовувати папір формату А3 (297×420 мм).

На останній сторінці пояснювальної записки необхідно наклеїти конверт, в який вкласти компакт-диск. Студенти, які мають непарний варіант, наклеюють конверт у верхній частині листа, а інші - у нижній. Це потрібно для компактного розташування пояснювальних записок у архіві.

Оптимальний обсяг **основної частини** роботи (без додатків) має складати 25-35 аркушів. Обсяг додатків жорстко не лімітується, але пропонується мінімальний об'єм у кількості 25 аркушів.

Форматування пояснювальної записки:

- поля: ліве, верхнє і нижнє не менше 20мм, праве не менше 10мм;
- шрифт: Times New Roman, 14 pt;
- міжрядковий інтервал – 1,5 pt;
- відступ першого рядка – 1,25 см;
- вирівнювання: назв розділів – по центру, назв підрозділів, пунктів і підпунктів – по ширині, основного тексту – по ширині.

Заголовки структурних частин курсової роботи: «ЗМІСТ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ», «ДОДАТКИ», «ПЕРЕЛІК ПОСИЛАНЬ» та назви розділів друкуються великими літерами. Кожну структурну частину роботи потрібно починати з нової сторінки.

4.2 Нумерація

Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, рисунків, таблиць, формул подають арабськими цифрами без знака «№».

Першою сторінкою курсової роботи є титульний аркуш, який включають до загальної нумерації, але номер сторінки на ньому не ставлять. На всіх наступних сторінках номер проставляють у правому верхньому куті сторінки без крапки в кінці.

Такі структурні частини роботи, як «ЗМІСТ», «ВСТУП», «ВИСНОВКИ», «ДОДАТКИ», «ПЕРЕЛІК ПОСИЛАНЬ» не мають порядкового номеру.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку. В кінці номера підрозділу крапка не ставиться. Потім у тому ж рядку наводять назву підрозділу.

Аналогічним чином нумерують пункти (порядкова нумерація в межах підрозділу) і підпункти (порядкова нумерація в межах пункту). Наприклад, «2.3.1» – перший пункт третього підрозділу другого розділу.

Переліки, за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку, або крапку в залежності від контексту.

Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (перший рівень деталізації).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації).

Приклад:

а) форма і розмір клітин;

б) живий склад клітин:

1) частини клітин;

2) неживі включення протопластів;

в) утворення тканини.

Переліки першого рівня деталізації друкують з абзацного відступу, другого рівня – звідступом відносно місця розташування переліків першого рівня.

Ілюстрації (фотографії, скріншоти, креслення, схеми, графіки, рисунки, карти) і таблиці необхідно подавати безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Ілюстрації позначають словом «Рисунок» і нумерують послідовно в межах розділу, за винятком ілюстрацій, поданих у додатках. Номер ілюстрації повинен складатися з номера розділу і порядкового номера ілюстрації, між якими ставиться крапка. Номер, назва і пояснювальний підпис (у разі необхідності) повинні міститися безпосередньо під ілюстрацією.

Таблиці нумерують послідовно в межах розділу (за винятком таблиць, поданих у додатках). В лівому верхньому куті над відповідним заголовком таблиці розміщують напис «Таблиця» із зазначенням її номера. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка. Якщо таблиця займає більше одного аркуша, її заголовок переноситься на наступний, в лівому верхньому куті над відповідним заголовком таблиці розміщують напис «Продовження таблиці» із зазначенням її номера, без назви таблиці.

Формули в курсовій роботі нумерують в межах розділу. Номер формули повинен складатися з номера розділу і порядкового номера формули, між якими ставиться крапка. Номер формули пишуть в круглих дужках і розміщують біля правого поля аркуша на рівні відповідної формули.

Посилання в тексті роботи на ілюстрації, таблиці, формули вказують порядковим номером в круглих дужках, наприклад, «... у формулі (2.1)».

4.3 Оформлення цитат і переліку посилань

Бібліографічний опис літературних джерел можна розміщувати одним із таких способів: у порядку появи посилань у тексті, в алфавітному порядку прізвищ перших авторів або заголовків, у хронологічному порядку.

Посилання в тексті роботи на літературні джерела вказують порядковим номером в квадратних дужках, наприклад, «... в роботі [4] показано ...». Якщо в курсовій роботі наводиться цитата, то обов'язковим є зазначення сторінок першоджерела, що містять цитовану інформацію, наприклад, «...[4, с. 10-12]». Цитати беруться в лапки і виділяються курсивом. Приклади бібліографічного опису окремих видів літературних джерел наведений в додатку Д.

4.4 Оформлення додатків

Додаток повинен мати заголовок, надрукований на початку сторінки із вирівнюванням по центру сторінки. На одну строку вище від заголовку друкується слово «Додаток ____» і велика літера, що позначає додаток. Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Ґ, Є, І, Ї, Й, О, Ч, Ї.

Текст кожного додатку за необхідності може бути поділений на розділи й підрозділи, які нумерують в межах кожного додатка. В цьому разі перед кожним номером ставлять позначення додатка (літеру) і крапку, наприклад А.2.1 – перший підрозділ другого розділу додатка А. Ілюстрації, таблиці і формули нумерують аналогічним чином.

5 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОЇ РОБОТИ

5.1 Хід виконання та захисту курсової роботи

Керівник здійснює контроль за ходом виконання студентом, надає йому необхідну консультативну допомогу.

Протягом семестру студент демонструє викладачу поточні результати роботи над проектом.

В терміни, визначені графіком виконання (таблиця 5.1), курсова робота у зброшурованому вигляді здається керівникові на перевірку. До пояснювальної записки додається електронний варіант на оптичному носії (прикріпленому до останньої сторінки обкладинки), оформлений згідно з розділу.

Таблиця 5.1 – Графік виконання курсової роботи

Термін (№ тижня)	Номер етапу	Назва етапів курсової роботи
5	1.	Отримання теми курсової роботи
7	2.	Підготовка ТЗ
9	3.	Пошук та вивчення літератури з питань курсової роботи
10	4.	Розробка алгоритму вирішення задачі
10	6.	Узгодження алгоритму з керівником
11	5.	Розробка сценарію роботи програми
11	6.	Узгодження сценарію роботи програми з керівником
11	7.	Узгодження з керівником інтерфейсу користувача
14	8.	Розробка програмного забезпечення
14	9.	Налагодження розрахункової частини програми
15	10.	Розробка та налагодження інтерфейсної частини програми
15	11.	Узгодження з керівником набору тестів для контрольного прикладу
16	12.	Тестування програми
17	13.	Підготовка пояснювальної записки
17	14.	Здача курсової роботи на перевірку
18	15.	Захист курсової роботи

Після перевірки курсової роботи викладач призначає день, час і місце захисту. До захисту курсової роботи допускаються студенти, які виконали всі вимоги навчальної програми та календарного плану, своєчасно представили роботу й усі необхідні матеріали.

Захист курсової роботи 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** даних вказівок, викладач виставляє студенту оцінку.

На оцінку за КР впливають:

- якість розробленого програмного забезпечення;
- якість розробленої програмної документації;
- компетентність та загальна ерудиція студента при відповідях на запитання під час захисту;
- ступінь виконання графіку підготовки курсової роботи.

Якщо студент подав на захист не самостійно виконану роботу, про що свідчить його некомпетентність у рішеннях та матеріалах роботи, КР до захисту не допускається, що супроводжується записом "не допущений" у екзаменаційній відомості. Такий самий запис робиться у випадку, якщо КР не завершена на час захисту. В цих випадках запис "не допущений" еквівалентний отриманню оцінки "незадовільно".

5.2 Критерії оцінювання курсової роботи

Рейтингова оцінка з курсової роботи має дві складові: виконання курсової роботи та її захист.

Перша (стартова) складова характеризує роботу студента з курсового проектування та її результат - якість пояснювальної записки та розробленого програмного забезпечення; друга складова характеризує якість захисту студентом курсової роботи.

Розмір шкали першої складової дорівнює 70 балів, а другої складової - 30 балів.

Система рейтингових балів.

Стартова складова виконання курсової роботи (r1):

- якість проведення об’єктно-орієнтованого аналізу предметної області задачі – до 5 балів;
- якість проектування класів – до 10 балів;
- повнота та коректність алгоритмізації методів класів – до 10 балів;
- якість проектування діаграми класів – до 5 балів;
- якість написання та оформлення програмного коду – до 10 балів;
- складність розробки інтерфейсу користувача (меню, екранні форми, графіки тощо) та графічного представлення результатів – до 10 балів;
- якість проведення тестування працездатності програмного забезпечення - до 5 балів;
- якість оформлення пояснювальної записки з урахуванням виконання вимог нормативних документів – до 15 балів;
- несвоєчасність виконання основних етапів графіку підготовки курсової роботи та відправки на перевірку – 1 бал пониження максимального балу за кожен день запізнення відповідно графіку.

Узагальнені критерії оцінювання виконання курсової роботи та деталізовані бальні шкали наведено в таблиці 5.2.

Таблиця 5.2 – Параметри та критерії оцінювання виконання курсової роботи

Параметри оцінювання	Діапазон балів	Критерії оцінювання за бальною шкалою
Якість проведення об’єктно-орієнтованого аналізу предметної області задачі	3-5	3 – реферативний опис предметної області виконано не повністю, словник предметної області сформований не достатньо якісно
		4 – реферативний опис предметної області виконано не повністю, однак словник предметної області сформований якісно
		5 – реферативний опис предметної області виконано повністю, словник предметної об-

		ласті сформований якісно
Якість проектування класів	6-10	6-9 - класи спроектовані недостатньо якісно
		10 - усі класи спроектовані якісно
Повнота та коректність алгоритмізації методів класів	6-10	6-9 - коректні алгоритми не усіх методів
		10 - алгоритми усіх методів коректні
Якість проектування діаграми класів	3-5	0 – діаграми класів та опис відсутні
		5 - діаграма класів коректна
Якість написання та оформлення програмного коду	6-10	2 - повне використання власних заголовних файлів
		2 – програмний код повністю документований (має коментарі)
		6..9 – тестова програма працює правильно, функціональність не усіх методів класів відповідає передбаченим алгоритмам (пропорційно кількості методів), програмний код повністю документований (має коментарі), має місце повне використання власних заголовних файлів; класи реалізовані у повній відповідності до методології ООП (пропорційно кількості класів)
		10 – тестова програма працює правильно, функціональність усіх методів класів відповідає передбаченим алгоритмам, програмний код повністю документований (має коментарі), має місце повне використання власних заголовних файлів; класи реалізовані у повній відповідності до методології ООП

Складність розробки інтерфейсу користувача (меню, екранні форми, графіки тощо) та графічного представлення результатів	6-10	6..9 – програмний додаток має віконний інтерфейс, включає меню, однак інтерактивні інтерфейсні засоби спроектовані недостатньо якісно (пропорційно кількості форм)
		10 – програмний додаток має гарно спроектований віконний інтерфейс, включає меню і інтерактивні інтерфейсні засоби (графічні екранної форми для введення даних та графічне представлення результатів)
Відповідність оформлення пояснювальної записки встановленим вимогам	9-15	9-14 - текст курсової роботи оформлено із порушеннями встановлених вимог
		15 - текст курсової роботи оформлено у відповідності до встановлених вимог
Не своєчасність виконання основних етапів графіку підготовки курсової роботи та відправки на перевірку	0 - $-\infty$	0 – курсова робота відправлена на перевірку у відповідності з графіком
		1 – пониження максимальної оцінки за 1 день затримки відправлення курсової роботи на перевірку згідно з графіком

Складова захисту курсової роботи (r2):

- ступінь володіння теоретичним матеріалом – до 5 балів;
- ступінь володіння алгоритмами програми в цілому – до 15 балів;
- вміння внести зміни у програмний код – до 10 балів.

Узагальнені критерії оцінювання захисту курсової роботи та деталізовані бальні шкали наведено в таблиці 5.3.

Таблиця 5.3 – Параметри та критерії оцінювання захисту курсової роботи

Параметри оцінювання	Діапазон балів	Критерії оцінювання за бальною шкалою
Ступінь володіння теоретичним матеріалом	0-5	0 – студент не дав відповіді на теоретичні питання

		1-4–відповідь, на одне чи два теоретичних питання
		5 - відповідь, на усі теоретичні питання
Ступінь володіння алгоритмами програми в цілому	0-15	0 – студент не володіє алгоритмами програми
		1-14 – студент частково володіє алгоритмами програми
		15 –студент повністю володіє алгоритмами програми
Вміння внести зміни у програмний код	0-10	0 – студент не може виконати додаткові завдання, пов’язані із внесенням змін у програмний код
		1-9 – студент частково може виконати додаткові завдання, пов’язані із внесенням змін у програмний код
		10 – студент грамотно виконує додаткові завдання, пов’язані із внесенням змін у програмний код

Сума балів двох складових переводиться до залікової оцінки згідно з таблицею 5.4.

Таблиця 5.4 –Загальний критерій оцінювання

$R = r1 + r2$	Оцінка ECTS	Національна оцінка
95... 100	A	відмінно
85 ... 94	B	добре
75 ... 84	C	
65 ... 74	D	задовільно
60 ... 64	E	
Менше 60	Fx	незадовільно
Курсовий проект не допущено до захисту	F	не допущено

6 ТЕМИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ НА КУРСОВУ РОБОТУ

Тематика курсових робіт визначається змістом робочої навчальної програми з дисципліни «ОП», що читається студентам денної форми навчання протягом 2-го семестру в обсязі 5 кредитів ECTS (в тому числі 36 годин лекцій та 54 годин комп'ютерного практикуму).

Студент повинен написати порівняно складну програму з використанням графічних можливостей мови програмування (з графічним інтерфейсом). Програма повинна відповідати всім критеріям якості програмного продукту. Для виконання курсової роботи використовується мова програмування C/C++ або інша, за погодженням з викладачем.

Порядок виконання курсової роботи включає розробку алгоритмічного і програмного забезпечення, проектування зручного інтерфейсу користувача та оформлення пояснювальної записки.

Студент може запропонувати власну тему, обґрунтувавши її актуальність та доцільність виконання відповідної розробки. При цьому він складає технічне завдання, яке затверджується керівником курсової роботи.

6.1 Варіанти завдань

6.1.1 Тематика базових індивідуальних завдань

Розробити програму у відповідності до індивідуального завдання (табл. 6.1). В програмі передбачити:

- 1) введення вихідних даних за допомогою розробленої самостійно екранної форми (введення частини даних організувати засобами меню);
- 2) збереження отриманих числових результатів у вигляді текстового файлу; (обрана мова програмування обов'язково має містити функції взаємодії з файловою системою);
- 3) за можливості графічне представлення одержаних результатів;
- 4) демонстрацію (за можливості - графічну) роботи застосованих методів (алгоритмів) розв'язання підзадач;
- 5) оцінку складності алгоритмів (аналітичну);

б) порівняльний аналіз методів (алгоритмів) розв'язання підзадач.

Таблиця 6.1 – Теми індивідуальних завдань на курсову роботу

	Тема	Вхідні дані
1.	Розв'язання СЛАР точними методами	Система лінійних рівнянь (загальний вигляд системи визначається керівником). Методи розв'язання: <ul style="list-style-type: none">– LUP-метод;– матричний метод;– метод Гауса-Холецького (Квадратного кореня).
2.	Упорядкування масивів	Масиви (розмір – не менше 100 елементів, генеруються випадковим чином, максимально до 50000). Методи сортування: <ul style="list-style-type: none">– метод сортування вибором;– метод пірамідального сортування;– метод плавного сортування.
3.	Розв'язання трансцендентних рівнянь	Трансцендентні рівняння (вказує керівник). Методи розв'язання: <ul style="list-style-type: none">– метод половинного ділення (метод бісекції);– метод дотичних (метод Ньютона);– метод січних.
4.	Пошук найкоротших шляхів в мережі	Граф (задає керівник). Методи пошуку: <ul style="list-style-type: none">– метод Флойда;– метод Данцига.
5.	Швидкісні методи множення матриць	Матриця (генерується випадковим чином). Розмірність має бути не менше 64*64. Методи розв'язання: <ul style="list-style-type: none">– метод Штрассена;

		<ul style="list-style-type: none"> – метод Винограда-Штрассена; – традиційний метод (для порівняння).
6.	Розв'язання СЛАР наближеними методами	<p>Система лінійних рівнянь (загальний вигляд системи визначається керівником).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод простої ітерації (Якобі); – метод Гауса-Зейделя; – метод найшвидшого спуску (градієнта).
7.	Упорядкування масивів	<p>Масиви (розмір – не менше 100 елементів, генеруються випадковим чином, максимально до 50000).</p> <p>Методи сортування:</p> <ul style="list-style-type: none"> – метод сортування злиттям (Д. фон Неймана); – метод швидкого сортування; – метод інтроспективного сортування.
8.	Зовнішнє упорядкування масивів	<p>Файл (розміром – не менше 10000000 цілочисельних елементів, генеруються випадковим чином).</p> <p>Методи сортування:</p> <ul style="list-style-type: none"> – метод природнього злиття; – метод збалансованого багатопрохідного злиття; – багатопрохідне сортування;
9.	Розв'язання нелінійних алгебраїчних рівнянь	<p>Нелінійні рівняння (вказує керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод половинного ділення (метод бісекції); – метод дотичних (метод Ньютона); – метод січних.
10.	Розв'язання за- дачі про вер-	<p>Граф (задає керівник).</p> <p>Методи розв'язання:</p>

	шинне покриття	<ul style="list-style-type: none"> – Жадібний метод; – метод Approx-Vertex-Cover.
11.	Розв'язання задач інтерполяції	<p>Система точок (вказує керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – лінійна інтерполяція; – метод Ньютона.
12.	Розв'язання задач інтерполяції	<p>Система точок (вказує керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – інтерполяція квадратичними сплайнами; – інтерполяція кубічними сплайнами.
13.	Розв'язання задач інтерполяції	<p>Система точок (вказує керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Лагранжа; – схема Ейткена.
14.	Розв'язання задач про найкоротший шлях	<p>Граф (задає керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Дейкстри; – метод Беллмана - Форда;
15.	Розв'язання СЛАР точними методами	<p>Система лінійних рівнянь (загальний вигляд системи визначається керівником).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Гауса (класичний); – метод Жордана-Гауса; – метод обертання.
16.	Пошук екстремумів функції однієї змінної	<p>Функція однієї змінної (погоджується з керівником), область пошуку, точність обчислень.</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод "золотого" перетину; – метод рівномірного пошуку;

		– діхотомії.
17.	Обернення матриці	Матриця (генерується випадковим чином або задає керівник). Методи розв'язання: – метод Жордана-Гауса; – метод Шульца.
18.	Розв'язання систем нелінійних рівнянь	Система нелінійних рівнянь (загальний вигляд системи визначається керівником). Методи розв'язання: – метод простої ітерації (Якобі); – метод Гауса-Зейделя;
19.	Обернення матриці	Матриця (генерується випадковим чином або задає керівник). Методи розв'язання: – метод Гауса (класичний) – метод LU-розкладу; – метод LUP-розкладу.
20.	Обернення матриці	Матриця (генерується випадковим чином або задає керівник). Методи розв'язання: – метод окаймлення; – метод розбиття на клітки.
21.	Знаходження найкоротшого остовного дерева	Граф (задає керівник) . Методи пошуку: – метод Прима; – метод Крускала; – метод Борувки.
22.	Розфарбовування	Граф (задає керівник).

	графів	<p>Методи розфарбовування:</p> <ul style="list-style-type: none"> – Жадібний метод; – Пошук з поверненням (MRV та степенева евристики).
23.	Пошук власних чисел та власних векторів матриці	<p>Матриця (задає керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Данилевського; – метод обертання.
24.	Пошук власних чисел та власних векторів матриці	<p>Матриця (задає керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Леверрье-Фадєєва. – метод А.Н. Крилова.
25.	Пошук заданих елементів у масиві	<p>Масиви (розмір – не менше 1000 унікальних елементів, генеруються випадковим чином). Шукане значення вводиться з клавіатури.</p> <p>Методи пошуку:</p> <ul style="list-style-type: none"> – послідовний метод (для порівняння); – бінарний метод; – однорідний бінарний метод; – метод Шара.
26.	Пошук заданих елементів у масиві	<p>Масиви (розмір – не менше 1000 унікальних елементів, генеруються випадковим чином). Шукане значення вводиться з клавіатури.</p> <p>Методи пошуку:</p> <ul style="list-style-type: none"> – послідовний метод (для порівняння); – метод Фібоначі; – інтерполяційний метод;

		– метод Хеш-функції.
27.	Розв'язання СЛАР точними методами	Система лінійних рівнянь (загальний вигляд системи визначається керівником). Методи розв'язання: – метод Крамера – метод Гауса з одиничною діагоналлю; – метод Гауса з вибором головного елемента.
28.	8-puzzle	Початкова розстановка (задає керівник). Методи розв'язання: – LDFS; – BFS;
29.	Чисельне обчислення інтегралів	Декілька типів інтегралів (вказує керівник). Методи розв'язання: – метод прямокутників; – метод трапецій; – метод Сімпсона.
30.	Задача розмі- щення ферзів	Початкова розстановка (задає керівник). Методи розв'язання: – RBFS; – A*.
31.	Розв'язання систем нелінійних рівнянь	Система нелінійних рівнянь (загальний вигляд системи визначається керівником). Методи розв'язання: – метод січних; – метод Ньютона.
32.	Задача розміщення ферзів	Початкова розстановка (задає керівник). Методи розв'язання: – LDFS;

		<ul style="list-style-type: none"> – BFS; – IDS.
33.	Розв'язання задач про знаходження шляху в лабіринті	<p>Граф (задає керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – метод Дейкстри; – метод А* (манхеттенська та евклідова евристики);
34.	Упорядкування масивів	<p>Масиви (розмір – не менше 100 елементів, генеруються випадковим чином, максимально до 50000).</p> <p>Методи сортування:</p> <ul style="list-style-type: none"> – метод Шелла (класичний); – метод Шелла (d = послідовність Седжвіка); – метод Шелла (d = послідовність Фібоначі); – метод Шелла (d = послідовність Токуда).
35.	Розв'язання задач про пошук кліки	<p>Граф (задає керівник).</p> <p>Методи розв'язання:</p> <ul style="list-style-type: none"> – Жадібний метод; – метод Брона – Кербоша.

6.1.2 Тематика додаткових індивідуальних завдань

1. Гра "Шахові фігури".

Необхідно розробити програму, яка розміщує на шахівниці фігури і перевіряє чи б'ють вони один одного.

Вихідними даними є текстовий файл з назвою і розташуванням фігур, наприклад:

king 0 0

queen 5 1

rook 1 3

bishop 3 7

knight 5 5

Підтримувані записи у вхідному файлі:

- king x y - король з координатами (x, y)
- queen x y - ферзь з координатами (x, y)
- rook x y - тура з координатами (x, y)
- bishop x y - слон з координатами (x, y)
- knight x y - кінь з координатами (x, y)

У вихідних даних можуть зустрічатися всі фігури крім пішаків. Кількість фігур у вихідних даних більше 1 і не перевищує 10. Фігури можуть повторюватися. В рамках задачі припускаємо, що координати фігур у вихідному файлі задані коректно (в межах дошки 8x8) і всі фігури розташовані в різних клітинах.

Необхідно створити всі фігури, намалювати розташування на екрані, щоб переглянути чи б'ють вони один одного, і вивести на екран список: яка фігура кого б'є.

2. Гра "Хрестики-нулики".

Розробити програму, яка дозволяє грати на необмеженому полі в «хрестики-нулики».

Мета гри: побудувати безперервну лінію з п'яти або більше фішок (хрестиків або нуликів) по горизонталі, вертикалі або діагоналі. Можливі два режими гри, коли виграє той, хто першим збудує лінію, або хто більше набере очок за певну кількість часу. У другому випадку кількість побудованих ліній не обмежена, кожна фішка в лінії приносить гравцеві одне очко.

Якщо в якості гравця виступає комп'ютер, програма робить перший хід. Роблячи черговий хід, програма аналізує ситуацію, розраховуючи можливі ходи супротивника вперед на 1-2 ходи, і в результаті проведеного аналізу ходить оптимальним чином.

Додаток повинен забезпечувати початок нової гри на чистому полі, а також перевірку відповідності дій гравців правилами гри та умови закінчення гри. Роль од-

ного з гравців (за вибором користувача) може виконувати комп'ютер. Для виключення можливості утворення ізолюваних ігор на одному полі і невинновданого збільшення розміру поля слід встановити максимально допустиму відстань (не більше п'яти) від вже існуючих фішок до нової, що розміщується гравцем.

3. Класичне судоку

Дано поле 9*9, розбите на квадрати 3*3 (рис. 6.1).

Рис. 6.1 – Поле класичного судоку

У кожен квадрат необхідно вписати цифри від 1 до 9 (без повторень). У кожному стовпці і в кожному рядку повинні бути всі цифри від 1 до 9 (рис. 6.2).

7	8	3	4	2	1	6	5	9
4	5	6	9	8	7	2	1	3
1	9	2	5	6	3	4	7	8
3	4	5	8	9	2	1	6	7
9	2	8	1	7	6	5	3	4
6	7	1	8	5	4	9	8	2
2	6	9	7	1	8	3	4	5
8	1	4	2	3	5	7	9	6
5	3	7	6	4	9	8	2	1

Рис. 6.2 – Приклад вирішеного судоку

У деяких клітинах спочатку дані цифри для єдиності рішення. Тестові завдання наведені на рис. 6.3–6.5.

1	2	5	6	7	8	3	9	4
4	3	8	5	9	2	1	6	7
7	6	9	3	4	1	2	5	8
8	5	2	4	3	6	7	1	9
6	4	7	1	8	9	5	3	2
3	9	1	2	5	7	8	4	6

5	1	2	4	7	9	3	8	6
4	8	9	6	3	5	7	1	2
7	3	6	2	1	8	4	5	9
8	9	7	1	4	2	6	3	5
1	5	4	9	6	3	8	2	7
2	6	3	5	8	7	9	4	1

	9		3	8				
						4		
7						6		
3	2							4
6			7		5			2
5							1	9

2	7	4	9	1	3	6	8	5
5	1	6	8	2	4	9	7	3
9	8	3	7	6	5	4	2	1

Рис. 6.3 – Тест №1

9	4	8	7	5	1	2	6	3
6	7	5	3	2	4	1	9	8
3	2	1	8	9	6	5	7	4

Рис. 6.4 – Тест №2

		4						8
		5						
				1	2		7	

Рис. 6.5 – Тест №3

4. Гра "Морський бій".

Дано поле 11*11 клітинок і набір кораблів:



Кожна палуба займає рівно 1 клітинку. Корабель може розташовуватися тільки по прямій лінії (горизонтально або вертикально). Кораблі не можуть стикатися навіть кутами. Знизу і праворуч від поля дано кількість палуб, які знаходяться в цьому ряду або стовпці. Частина полів може бути заповнена - поле позначено як пусте або містить частину корабля (окремо може бути зазначено ніс / корма, що дає інформацію про орієнтацію корабля в просторі і мінімальну кількість палуб, або внутрішня палуба) (рис. 6.6).

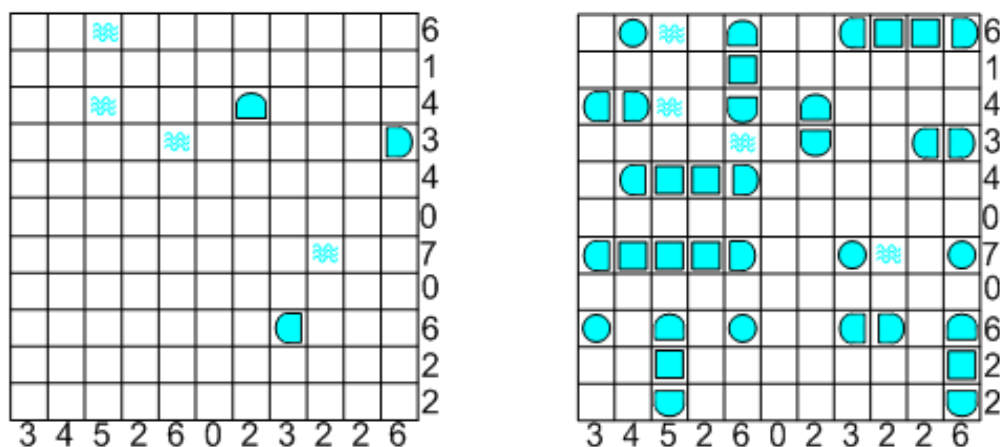


Рис. 6.6 – Поле гри «Морський бій» і приклад рішення

Розробити програму для гри в морський бій гравця з комп'ютером. Програма повинна дозволяти розставляти кораблі на ігровому полі, контролювати правильність їх розстановки, давати противникам можливість по чергово робити ходи і видавати відповідні інформаційні повідомлення. Коли в якості одного з гравців виступає

комп'ютер, програма повинна аналізувати попередні ходи і наступний робити на основі проведеного аналізу.

Ігрові об'єкти повинні бути реалізовані через класи.

Тестові завдання наведені на рис. 6.7–6.8.

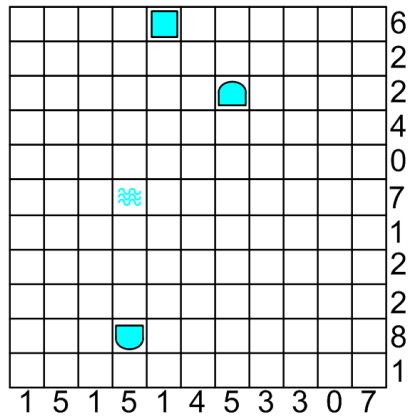


Рис. 6.7 – Тест № 1

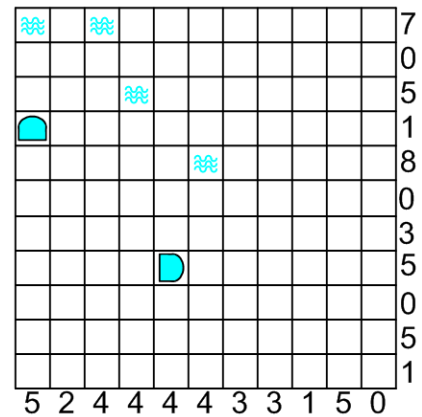


Рис. 6.8 – Тест № 2

5. Пентагон

Дано поле 12 * 12 (рис. 6.9) і набір фігур пентамімо (фігури, складені з 5 квадратів, див. рис 6.10).

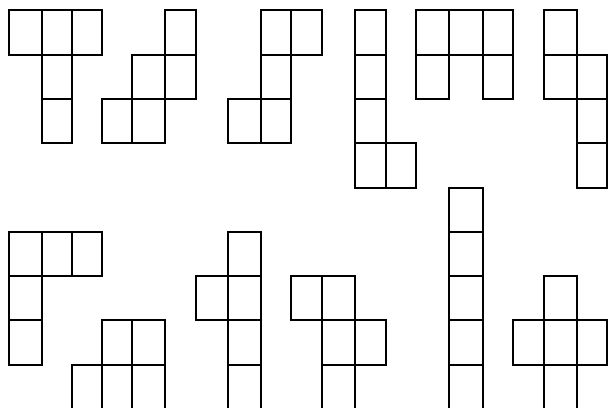
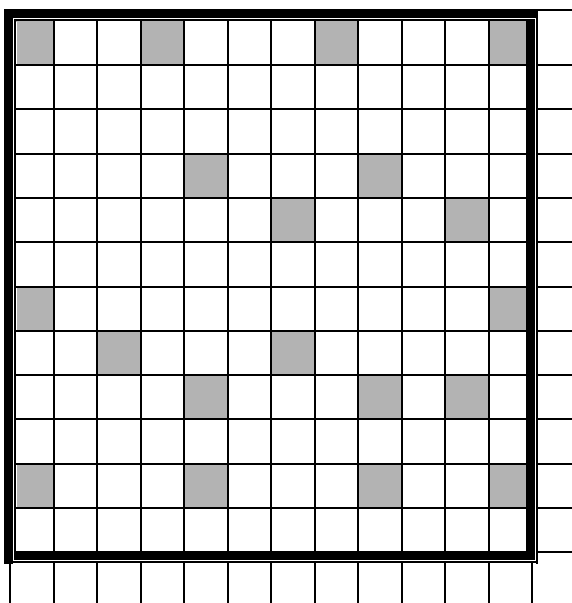


Рис. 6.9 – Приклад ігрового поля

Рис. 6.10 – Набір фігур пентамімо

Необхідно розташувати фігури всередині поля таким чином, щоб вони не стикалися навіть кутами. Фігури пентамімо можна повертати, але не можна дзеркально відображати. Частина порожніх клітин поля може бути вказана. Цифри праворуч і ліворуч від поля показують, скільки клітин пентамімо розташовано в даному ряду або стовпці.

Тестові завдання наведені на рис. 6.11-6.12.

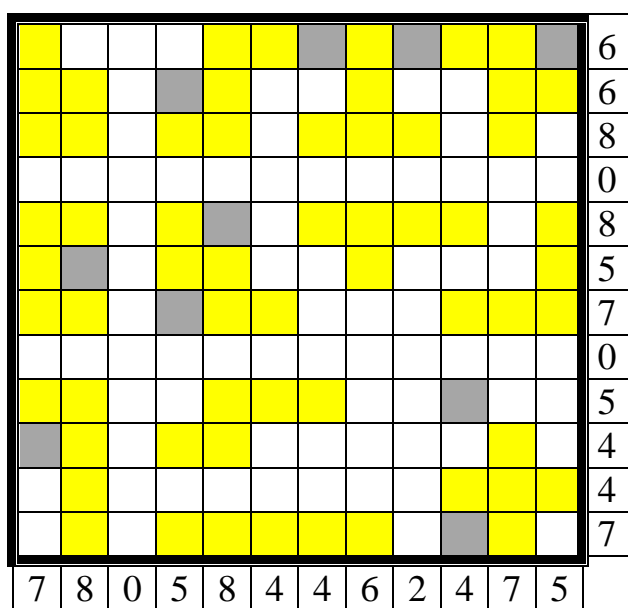


Рис. 6.11 – Тест № 1

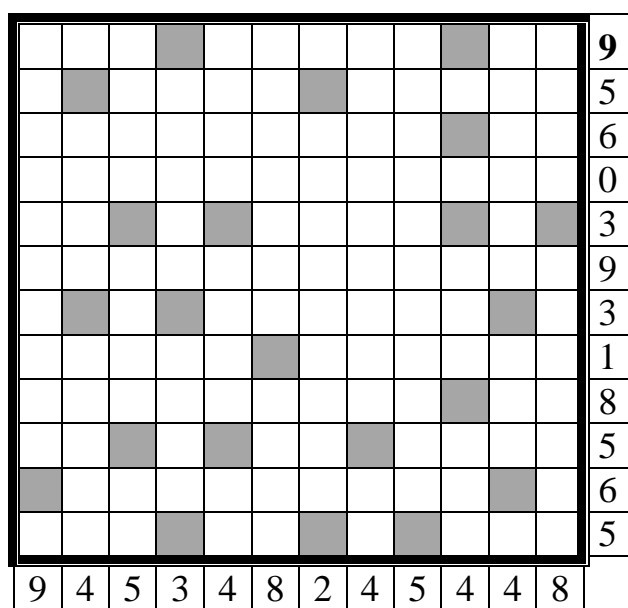


Рис. 6.12 – Тест № 2

6. Стрілки

Дано поле 8*8. Зверху, знизу зліва і справа дані поля, в яких треба занести стрілки, що вказують всередину квадрата (рис. 6.13).

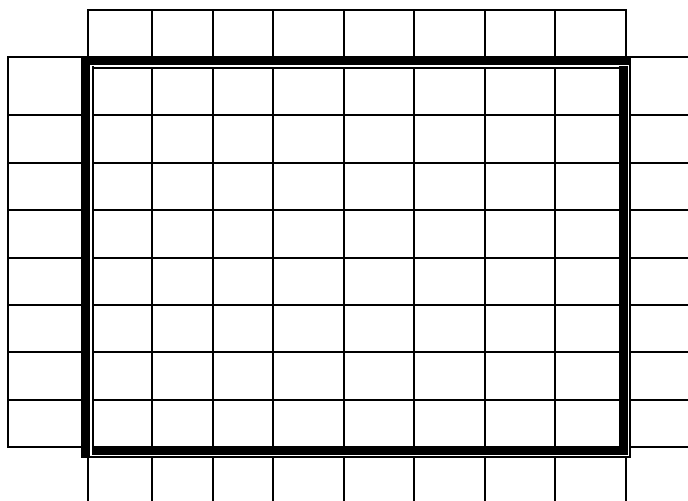


Рис. 6.13 – Приклад ігрового поля «Стрілки»

Усередині квадрата розташовані цифри, які вказують, скільки стрілок направлено на дане поле (рис. 6.14).

	↓	↘	↓	
→	5	2	5	←
↗	3	0	1	↖
→	4	3	4	←
	↑	↗	↖	

Рис. 6.14 – Приклад решеного поля «Стрелки» 3*3

На підставі цифр усередині необхідно розставити стрілки.

Тестові завдання рис. 6.15-6.16.

	3	5	3	4	3	3	4	5
	3	3	4	3	5	2	3	6
	0	1	1	3	1	1	2	2
	2	1	2	1	2	1	2	3
	2	5	2	2	3	2	3	5
	3	3	4	3	2	2	4	5
	1	3	4	4	3	2	4	4
	2	4	4	5	6	4	3	4

Рис. 6.15 – Тест № 1

	5	5	6	3	2	3	5	2
	4	3	4	2	2	2	3	2
	6	3	3	4	4	3	3	4
	1	1	2	1	2	1	3	0
	4	2	5	3	2	5	4	4
	3	3	4	4	3	1	5	2
	2	1	2	1	1	2	1	3
	3	1	4	2	2	2	4	1

Рис. 6.16 – Тест № 2

7. Вказівники

Дано поле 5*5. Усередині поля дані вказівники, у напрямку яких можна рухатися. Необхідно, стартувавши з клітки з номером 1 (верхній лівий кут) дістатися до клітки з номером 25, яка розташована у лівому нижньому кутку, побувавши в кожній клітині рівно один раз (рис. 6.17).

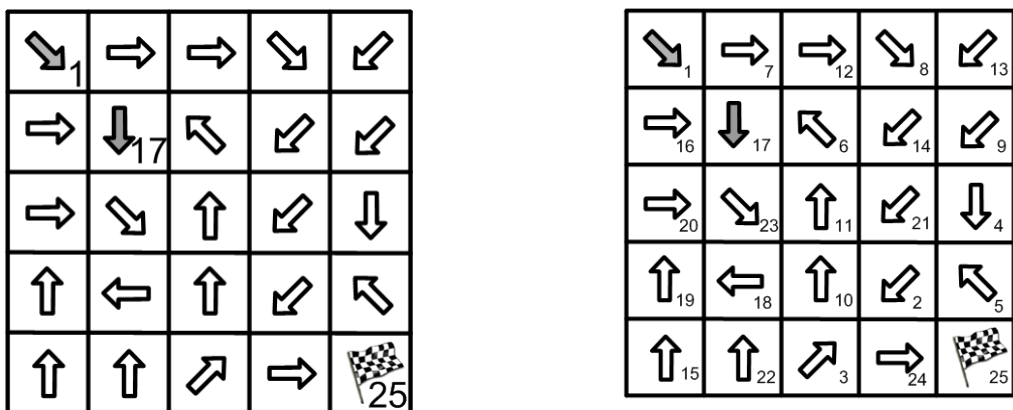


Рис. 6.17 – Ігрове поле «Вказівники» і приклад рішення

Тестові завдання - рис. 6.18-20.

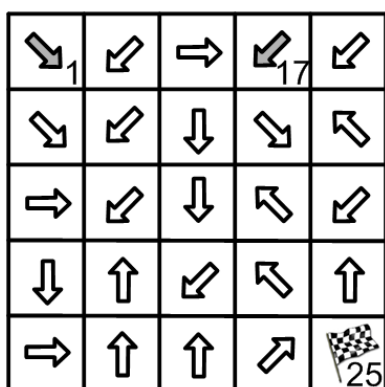


Рис. 6.18 – Тест № 1

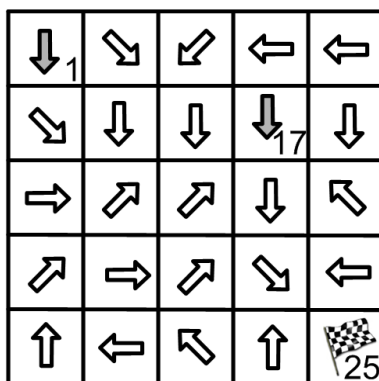


Рис. 6.19 – Тест № 2

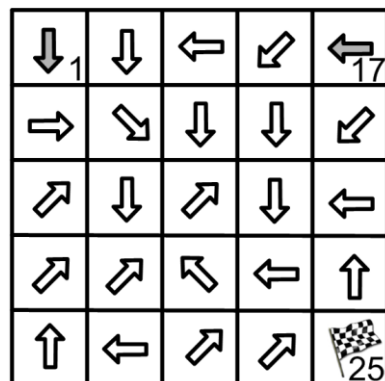


Рис. 6.20 – Тест № 3

8. Розміщення

Дано поле 15 * 15. Зліва, справа, знизу і зверху вказані числа. Числа зліва і зверху показують, який найбільш довгий блок чорних клітин присутній в цьому рядку (стовпці). Аналогічно - числа праворуч і знизу вказують, який найбільш довгий блок з білих (жовтих) клітин присутній в даному стовпці або рядку (рис. 6.21). Таких блоків може бути кілька.

	2	5	6	5	2	
1						1
5						0
3						1
3						1
3						1
5						0
	3	1	0	1	3	

	2	5	6	5	2	
1						1
5						0
3						1
3						1
3						1
5						0
	3	1	0	1	3	

Рис. 6.21 – Ігрове поле «Розміщення» і приклад рішення

Тестові завдання - рис. 6.22-23.

	7	7	3	3	2	1	4	3	1	7	4	2	4	6	4	
5																6
6																4
3																10
2																7
2																11
2																10
3																4
2																13
6																6
5																5
7																5
6																3
2																6
2																7
4																8
	4	4	6	7	7	8	5	10	10	5	6	5	3	3	5	

Рис. 6.22 – Тест №1 (№44-2011)

	3	4	8	3	1	7	6	1	1	10	1	1	1	7	4	
5																6
1																5
2																4
2																5
2																8
5																3
1																3
7																2
1																3
10																3
1																7
6																7
2																5
1																5
6																8
	9	9	5	6	8	5	3	4	5	5	7	4	3	5	10	

Рис. 6.23 – Тест №2 (№52-2010)

9. Доміно

Розмістіть в сітці комплект елементів доміно від 12 до 56 без дублів. Елементи (рис. 6.24) не повинні торкатися один одного навіть кутом. У рядках і стовпчиках цифри не повинні повторюватися. Числа, що стоять у сітці, означають суму цифр з доміно, що стикаються із відповідною кліткою стороною або кутом. У клітці з числом доміно знаходитися не може (рис. 6.25).

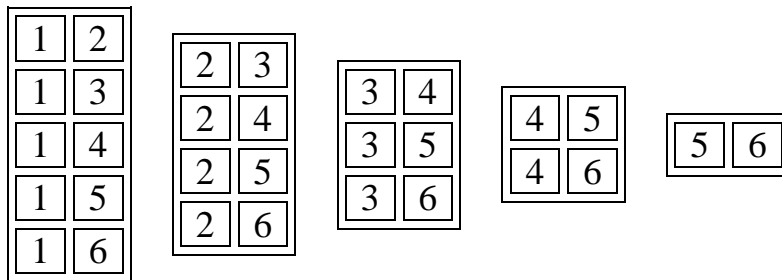


Рис. 6.24 – Елементи доміно

	14			14		11		
		10	10			14	8	8
6	13		12		12			
	14					5	10	10
		6		17	17			
12		4	6				10	10
		6		13	15			
	11							
		11		8	6		9	3

6	14	4	5	14	3	11	2	1
4		10	10		6	14	8	8
6	13	1	12		12		3	2
2	14	6		5	1	5	10	10
5		6		17	17		1	4
12		4	6	6	5		10	10
3	4	6		13	15			5
	11		2			4		3
1	3	11	6	8	6	2	9	3

Рис. 6.25 – Ігрове поле «Доміно» і приклад рішення

Тестові завдання - рис. 6.26-6.27.

		11				9	10	10
9		13						
8		12	7	8	9		17	
8		9			12		14	
17		16	9		13			
					5	5	14	10
	20		11		7	7		
				11				14
	10	6	6	11		11		

			12					2
			12			8	11	5
14	16	6	7	7	10	6	13	
				5				
		6		11	12	13		
	6	6		16		12		
	6			14		16	10	10
7	12			15				7
5								7

Рис. 6.26-6.27 – Тест №1, 2

10. Хмарочоси

Суть головоломки КВАРТАЛ (або ХМАРОЧОСИ): на полі $N \times N$ повинні розміститися будівлі різної висоти таким чином, щоб в кожному вертикальному і кожному горизонтальному ряду були присутні будинку тільки різної поверховості (від 1 до N поверхів, будинки позначаються числами, рівними їх поверховості). Числа поза діаграми показують, скільки будинків видно спостерігачеві, якщо він знаходиться на місці даного числа (див. Приклад №1 і №2 на рис. 6.28-6.29).

			2		

Приклад №1

			2		

			3		

Приклад №2

			3		

Рис. 6.28-6.29 – Ігрові поля «Хмарочоси» і приклад рішення ($N=4$)

Тестові завдання - рис. 6.30-6.31.

	1	2	2	2	4	
1						3
2						2
3						2
4						2
3						1
	4	4	2	2	1	

					3	
3						
3						3
						3
	3			3		

Рис. 6.30 – Тест №1, 2 ($N=5$)

	3	5	2	2	1	2	
4							2
2							2
4							1
3							2
1							4
2							3
	2	1	4	3	2	3	

	1	2	5	4	4	3	4	
1								4
2								3
5								2
4								3
4								1
2								4
3								3
	3	4	2	4	1	3	2	

Рис. 6.31 – Тест №3, 4 ($N=6,7$)

11. Гра у 15

Гра складається з 15 однакових квадратних пластинок з нанесеними числами від 1 до 15. Пластинки поміщаються в квадратну коробку, довжина сторони якої в чотири рази більша довжини сторони пластинок, відповідно в коробці залишається незаповненим одне квадратне поле. Мета гри — переміщуючи пластинки по коробці добитися впорядкування їх по номерах (як зображено на рисунку 6.32), бажано зробивши якомога менше переміщень.



Рис. 6.32

12. Варіації карткових ігор (таблиця 6.1)

Таблиця 6.1

№	Гра (мінімум на 4 гравці)
1	101 https://www.durbetsel.ru/2_101.htm
2	Курка https://www.durbetsel.ru/2_kurisa.htm
3	Чотири рядки https://www.durbetsel.ru/2_chetyre_stroki.htm
4	Выбух https://www.durbetsel.ru/2_vzryv.htm
5	Останній гравець https://www.durbetsel.ru/2_posledny_igrok.htm
6	Blackjack https://en.wikipedia.org/wiki/Blackjack
7	Редуду https://www.durbetsel.ru/2_redudu.htm
8	Эльферн https://www.durbetsel.ru/2_elfern.htm

СПИСОК ЛИТЕРАТУРЫ

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ, 2001
2. Майника Э. Алгоритмы оптимизации на сетях и графах, - М.: Мир, 1981
3. Окулов С. М. Программирование в алгоритмах. — М.: БИНОМ. 2002
4. Д.Э. Кнут. Искусство программирования, том 3. Пер. с англ. М.: Издательский дом "Вильямс", 2007. (и др. издания)
5. Буч Г., Объектно-ориентированный анализ и проектирование с примерами приложений. Издательство Вильямс , С.-Петербург, 2008.
6. Бертран Мейер. Объектно-ориентированное конструирование программных систем. — М.: Русская редакция, 2005.
7. Мацяшек Лешек А. Анализ и проектирование информационных систем с помощью UML 2.0, 3-е изд.: — М.: ООО «И.Д.Вильямс», 2008.
8. Лафоре Р. Объектно-ориентированное программирование в C++. С.-Петербург: Питер, 2006.
9. Троелсен Э. Язык программирования C# 2008 и платформа .NET 3.5. Издательство: Вильямс , С.-Петербург, 2010.
10. Синтес А. Освой самостоятельно объектно-ориентированное программирование за 21 день. Москва; С.-Петербург: Вильямс, 2002.
11. Шмуллер, Джозеф. Освой самостоятельно UML за 24 часа. - 3-е изд. М.: Вильямс , 2005

ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра
автоматизованих систем обробки інформації та управління

Затвердив

Керівник _____

« ____ » _____ 201_ р.

Виконавець:

Студент _____ <ПІБ> _____

« ____ » _____ 201_ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

на тему: <Тема курсової роботи>

з дисципліни:

«Об'єктно-орієнтоване програмування»

1. *Мета:* Метою курсової роботи є розробка
2. *Дата початку роботи:* « ____ » _____ 202_ р.
3. *Дата закінчення роботи:* « ____ » _____ 202_ р.
4. *Вимоги до програмного забезпечення.*
 - 1) Функціональні вимоги:
 - Можливість
 - < вимоги до функціональних характеристик >.
 - 2) Нефункціональні вимоги:
 - Можливість
 - < вимоги до надійності, складу і параметрів технічних засобів тощо >
 - Все програмне забезпечення та супроводжуюча технічна документація повинні задовольняти наступним ДЕСТам:
 - ГОСТ 29.401 - 78 - Текст програми. Вимоги до змісту та оформлення.
 - ГОСТ 19.106 - 78 - Вимоги до програмної документації.
 - ГОСТ 7.1 - 84 та ДСТУ 3008 - 95 - Розробка технічної документації.
5. *Стадії та етапи розробки:*
 - 1) Об'єктно-орієнтований аналіз предметної області задачі (до __.__.202_ р.)
 - 2) Об'єктно-орієнтоване проектування архітектури програмної системи (до __.__.202_ р.)
 - 3) Розробка програмного забезпечення (до __.__.202_ р.)
 - 4) Тестування розробленої програми (до __.__.202_ р.)
 - 5) Розробка пояснювальної записки (до __.__.202_ р.).
 - 6) Захист курсової роботи (до __.__.202_ р.).
6. *Порядок контролю та приймання.* Поточні результати роботи над КР регулярно демонструються викладачу. Своєчасність виконання основних етапів графіку підготовки роботи впливає на оцінку за КР відповідно до критеріїв оцінювання.

ДОДАТОК Б ТИТУЛЬНА СТОРІНКА

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

(повна назва кафедри, циклової комісії)

КУРСОВА РОБОТА

з _____

(назва дисципліни)

на тему: _____

Студента (ки) _____ курсу _____ групи

Спеціальності 121 «Інженерія програмного забезпечення»

Керівник _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Кількість балів: _____

Національна оцінка _____

Члени комісії

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

Київ- 202_рік

ДОДАТОК В ЛИСТ ЗАВДАННЯ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Об'єктно-орієнтоване програмування

Спеціальність "Інженерія програмного забезпечення"

Курс 1 Група ІІ-XX

Семестр 2

ЗАВДАННЯ

на курсову роботу студента

(прізвище, ім'я, по батькові)

1. Тема роботи _____

2. Строк здачі студентом закінченої роботи _____

3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1.	Отримання теми курсової роботи		
2.	Підготовка ТЗ		
3.	Аналіз предметної області		
4.	Проектування архітектури програмної системи		
5.	Розробка сценарію роботи програми		
6.	Узгодження з керівником інтерфейсу користувача		
7.	Розробка програмного забезпечення		
8.	Узгодження з керівником плану тестування		
9.	Тестування програми		
10.	Підготовка пояснювальної записки		
11.	Здача курсової роботи на перевірку		
12.	Захист курсової роботи		

Студент

(підпис)

(прізвище, ім'я, по батькові)

Керівник

(підпис)

(прізвище, ім'я, по батькові)

"__" _____ 202__ р.

ДОДАТОК Г ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду програмного забезпечення

<Тема курсової роботи>

(Найменування програми (документа))

CD-RW

(Вид носія даних)

арк, Кб

(Обсяг програми (документа), арк., Кб)

студента I курсу групи ІІІ-ХХ

<ІІІБ>