

**ФАХОВИЙ КОЛЕДЖ
ЗАКЛАДУ ВИЩОЇ ОСВІТИ «УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

Циклова комісія з інформаційних технологій

МЕТОДИЧНІ ВКАЗІВКИ
щодо виконання курсової роботи з дисципліни
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»
для здобувачів фахової передвищої освіти
денної та заочної форми навчання

Освітньо-професійна програма «Інженерія програмного забезпечення»
Спеціальність 121 Інженерія програмного забезпечення
Галузь знань 12 Інформаційні технології

Затверджено Педагогічною радою Фахового коледжу
Протокол № 1 від «29» серпня 2022 р.
Голова Педагогічної ради



Інна Варварук
Інна ВАРВАРУК

Схвалено методичною радою Фахового коледжу
ЗВО «Університет Короля Данила»
Протокол №1 від «26» серпня 2022 р.
Голова методичної ради

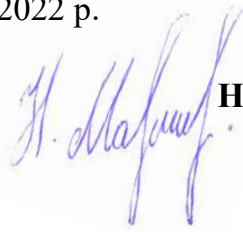
Тетяна Зуб'як

Тетяна ЗУБ'ЯК

Методичні вказівки щодо виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» розглянуто і схвалено на засіданні циклової комісії з інформаційних технологій.

Протокол № 1 від «25» серпня 2022 р.

Голова циклової комісії



Наталія МАЛИНОВСЬКА

РОЗРОБНИК:

ПАШКЕВИЧ Олег Петрович – кандидат технічних наук.

ЗМІСТ

ВСТУП	4
1. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ	5
1.1. Порядок вибору та закріплення теми курсової роботи	5
1.2. Характеристика структури курсової роботи	5
1.3. Правила оформлення курсової роботи	8
2. ЗАХИСТ ТА КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ	11
3. ДОТРИМАННЯ НОРМ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ	11
4. ТЕМИ КУРСОВИХ РОБІТ	12
5. ВАРІАНТИ ЗАВДАНЬ ДО КУРСОВОЇ РОБОТИ	13
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	24
ДОДАТКИ	25

ВСТУП

Курсова робота є самостійною, практично-дослідницькою роботою, у процесі написання якої студент виявляє своє вміння відбирати, систематизувати і творчо осмислювати технічну документацію, працювати із спеціальною літературою, робити правильні і обґрунтовані висновки, знаходити в них головне, формулювати та логічно викладати свої думки.

Згідно з навчальним планом у 6 семестрі для здобувачів фахової передвищої освіти за освітньо-професійною програмою «Інженерія програмного забезпечення» передбачена курсова роботи з дисципліни «Об'єктно-орієнтоване програмування», яка є завершальним етапом вивчення відповідної дисципліни.

Мета написання курсової роботи – поглибити теоретичні знання, набуті студентами у процесі вивчення дисципліни і виробити вміння застосувати їх у практичній діяльності.

Завдання курсової роботи – навчити студентів користуватися та опрацьовувати технічну літературу, сприяти поглибленому засвоєнню набутих знань студентами з дисципліни «Об'єктно-орієнтоване програмування», ґрунтовному вивченню обраної мови програмування та популярних інструментів розробки.

Виконання курсової роботи повинно сприяти формуванню у студента навичок творчої та дослідницької роботи.

Тематика курсової роботи має відповідати завданням навчальної дисципліни і тісно пов'язуватися з практичними потребами конкретного фаху. Тематика розробляється викладачем – керівником роботи, погоджується на засіданні циклової комісії та затверджується директором Коледжу.

Захист курсової роботи проводиться публічно перед комісією у складі трьох осіб, у яку входять викладачі циклової комісії, заступник директора Коледжу, за участю керівника курсової роботи.

Здобувач освіти, який без поважної причини не подав курсову роботу в зазначений термін або не захистив її, вважається таким, що має академічну заборгованість. При отриманні незадовільної оцінки здобувач освіти за рішенням комісії виконує курсову роботу за новою тематикою або перепрацьовує попередню роботу в термін, визначений головою циклової комісії.

Курсові роботи повинні буди здані голові циклової комісії (для зберігання) не пізніше, ніж через два тижні після завершення курсового проектування. Відповідальність за своєчасну здачу курсових робіт несе безпосередньо керівник курсової роботи.

Курсові роботи зберігаються в архіві закладу освіти упродовж трьох років, а потім списуються в установленому порядку.

Професійні компетентності стандарту фахової передвищої освіти за спеціальністю 121 Інженерія програмного забезпечення (затвердженого наказом МОН України № 1006 від 21.09.2021 року), набуття яких забезпечується написанням курсової роботи

Код та назва компетентності	Результати навчання
СК 01. Здатність алгоритмічно та логічно мислити.	РН 02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного
СК 03. Здатність застосовувати	

теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.	забезпечення. РН 03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення.
<p>СК 04. Здатність дотримуватися стандартів при розробці програмного забезпечення.</p> <p>СК 07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.</p> <p>СК 09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.</p>	<p>РН 04. Використовувати знання математичних методів на рівні, необхідному для розв'язання типових задач програмної інженерії.</p> <p>РН 06. Використовувати основні методології та підходи до організації життєвого циклу програмного забезпечення.</p> <p>РН 08. Аналізувати вимоги до програмного забезпечення.</p> <p>РН 10. Обирати та застосовувати ефективні методи оптимізації алгоритмів.</p> <p>РН 15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.</p>

1. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ НАПИСАННЯ КУРСОВИХ РОБІТ

1.1. Порядок вибору та закріплення теми курсової роботи

Правильний вибір теми сприяє успішному написанню курсової роботи. Тема має бути добре знайомою, сучасною та актуальною, присвяченою розгляду важливих у теоретичному і практичному плані питань в контексті об'єктно-орієнтованого програмування.

Обов'язковим правилом вибору теми є те, що йому має передувати належна теоретична підготовка, тобто відповідне оволодіння необхідним матеріалом з предмету даного дослідження

Обирати самостійно складні для дослідження, багатоаспектні або широкі за обсягом теми не рекомендується, бо за обсягом часу, навчальної зайнятості та іншими об'єктивними причинами це може привести до поверхового викладу матеріалу, відсутності дослідження та недосягнення мети курсової роботи.

Здобувач освіти обирає тему курсової роботи на початку навчального семестру (Додаток А) зі списку тем, затверджених директором Фахового коледжу. Здобувач може запропонувати власну тему з обґрунтуванням доцільності її дослідження та узгодженням з науковим керівником.

Після обрання теми студент подає заяву на ім'я голови циклової комісії з юридичних дисциплін згідно з Додатком 1.

1.2. Характеристика структури та змісту курсової роботи

У вирішенні задачі курсової роботи потрібно виділяти наступні основні етапи, кожен з яких, у свою чергу, складається з множини взаємозв'язаних задач:

- постановка задачі;
- проектування програми;
- програмування;

- налагодження та тестування програми;
- оформлення пояснювальної записки та захист проекту.

Основним змістом першого етапу є формулювання призначення програми та поставка цілей, а також виявлення оригінальних задач, які будуть вирішуватися в процесі виконанні роботи. У зв'язку з цим виконання етапу потрібно почати із з'ясування (уточнення) основних понять і визначень, що зустрічаються в завданні і даній предметній області та огляду наявних аналогів. Для цих цілей може бути використана рекомендована література та інші джерела.

Після завершення аналітичного дослідження можна переходити до проектування програми. При цьому необхідно визначити структуру вхідних та вихідних даних, структуру функціональних модулів та взаємозв'язків між ними, розробити алгоритми роботи цих модулів та інтерфейс користувача.

На етапі програмування відбувається реалізація запропонованих архітектурних рішень у вигляді програмного коду. Варто зазначити, що крім формальної реалізації алгоритмів необхідно приділяти увагу стилю, дотриманню загально прийнятих та рекомендованих норм та правил найменування функціональних елементів та оформлення програмного коду загалом. Особливу увагу необхідно приділити розробці інтерфейсу користувача.

Наступним етапом розробки є тестування програми та детальний опис структури головного, допоміжного меню та діалогових вікон, а також рекомендації по використанню програми.

Оформлення пояснювальної записки є останнім етапом виконання курсового проекту і має за мету надати студентові навичок документування програмного продукту. Документування є завершальним етапом створення програмного виробу для курсової роботи. Вимоги до оформлення пояснювальної записки наведені в наступному розділі.

Структура курсової роботи

Курсова робота повинна містити:

1. Титульний аркуш.
2. Завдання на курсову роботу та календарний план її виконання.
3. Вступ.
4. Основну частину курсової роботи.
5. Висновки.
6. Список використаних джерел (не менше 25, включаючи нормативні акти).
7. Додатки (за необхідності).

Вимоги до змісту курсової роботи. Після затвердження теми курсової роботи студент за погодженням з науковим керівником розробляє план роботи, підбирає список літератури і опрацьовує необхідні нормативно-правові акти, а також визначає строки і складає орієнтований графік написання курсової роботи.

Курсова робота повинна бути написана відповідно до плану, який розробляється студентом. При роботі над обраною темою, студенту необхідно спочатку вивчити підібрані джерела. При цьому доцільно робити виписки у вигляді цитат або вільного переказу окремих положень. Після вивчення необхідної літератури, зібравши достатній матеріал для висвітлення теми, слід приступити до його повної обробки, упорядкування, аналізу і викладу згідно зі складеним планом. Студент повинен виразно, розбірливо, ясно, зрозуміло та логічно висловлювати свої та запозичені з використаних джерел думки. Необхідно, щоб автор при виконанні роботи висловлював власне ставлення до описуваних подій та оцінок їх у зібраній літературі. Тобто при написанні курсової роботи потрібно уникати поверхневого висвітлення, загальних фраз, а особливо, дослівного переписування з використаних

джерел. Останнє призведе до негативної оцінки роботи викладачем і повернення її на доопрацювання.

Курсова робота має бути акуратною, грамотною, а основні її положення та ідеї мають бути чітко і правильно сформульовані. Структурні елементи курсової роботи повинні відповідати встановленим вимогам:

Титульний аркуш курсової роботи містить назву закладу освіти, назву циклової комісії, де виконана робота; тему курсової роботи; прізвище, ім'я, по батькові автора; прізвище, ім'я, по батькові наукового керівника, його вчене звання, наукову ступінь; місто і рік (**додаток А**).

Зміст подають на початку курсової роботи, він містить найменування усіх розділів, підрозділів та пунктів, підпунктів (якщо вони мають заголовки), зокрема: вступу, розділів, підрозділів, загальних висновків, списку використаних джерел і додатків.

Перелік умовних позначень, символів, одиниць, скорочень і термінів, а також використані маловідомі скорочення, нові символи, позначення та їх перелік може бути поданий у курсовій роботі у вигляді окремого списку, який розміщується перед вступом. Перелік треба друкувати двома колонками, у яких зліва за абеткою наводять, наприклад, скорочення, а з права – їх детальне розшифрування. Якщо в курсовій роботі спеціальні терміни, позначення, скорочення повторюються менше трьох разів – перелік не складають, а їх розшифрування наводять у тексті при першому згадуванні.

У вступі коротко викладають оцінку сучасного стану проблеми, актуальність цієї роботи і підстави для її проведення, ціль роботи й галузь застосування.

Основна частина складається з розділів і підрозділів (пунктів і підпунктів). Кожен розділ повинен починатися з нової сторінки.

У першому розділі дається аналітичний огляд можливостей побудови систем, що вирішують поставлену задачу. Розглядаються особливості об'єктно-орієнтованого підходу і мов, що підтримують ООП. Наводиться обґрунтування обраного програмного забезпечення.

Другий розділ – проектний. У цьому розділі можна виділити 2–3 відносно самостійних параграфи, що розкривають теоретичні та практичні аспекти розробки програмного забезпечення.

З'ясовується семантика класів і об'єктів, визначається поведінку і атрибути кожної абстракції, визначається структура вхідних та вихідних даних, структура функціональних модулів та взаємозв'язки між ними, розробляється інтерфейс користувача.

Розділ повинен бути максимально насичений фактичною інформацією (таблиці, схеми, UML-діаграми) та всебічно відображати аспекти діяльності програмного забезпечення, що розробляється.

Третій розділ присвячений реалізації рішень, запропонованих та спроектованих у попередньому розділі. Структурно він вміщує 2 – 3 параграфи, які містять детальний опис роботи всіх блоків програмного забезпечення, надається опис отриманих результатів і інструкція з використання розробленої програми.

У висновках викладають найбільш важливі практичні результати, одержані в процесі виконання курсової роботи. У висновках необхідно акцентувати на якісних та кількісних показниках здобутих результатів, зокрема на архітектурних рішеннях., які було використано, зазначити, які функціональні вимоги вдалось реалізувати, оцінити швидкодію та обчислювальну складність реалізованих алгоритмів.

1.3. Правила оформлення курсової роботи

Загальні принципи оформлення

Як правило, курсова робота спочатку виконується в чорновому варіанті. Лише після узгодження з керівником дискусійних і сумнівних моментів, врахування зауважень і усунення недоліків, удосконалення структури викладу матеріалу приступають до чистового варіанту. Чистовий варіант курсової роботи потрібно акуратно підшити у папку.

Курсова робота виконується державною (українською) мовою на стандартному папері формату А4 (210×297 мм) білого кольору до тридцяти рядків на сторінці. При наборі використовують шрифт Times New Roman, кегль 14, міжрядковий інтервал 1,5.

Текст необхідно друкувати, залишаючи поля таких розмірів: ліве – не менше 25 мм, праве – не менше 10 мм, верхнє – не менше 20 мм, нижнє – не менше 20 мм.

Обсяг основного тексту курсової роботи повинен становити не більше 25–30 сторінок (без врахування таблиць та рисунків).

Кількість використаних джерел у курсовій роботі повинна складати не менше 25 найменувань.

У викладі матеріалу важливо дотримуватись принципу пропорційності, який полягає у дотриманні пропорцій між обсягами окремих частин роботи. При цьому на вступ відводиться 1–2 сторінки, на основну частину роботи 25–30 сторінок, на титульний лист і зміст роботи – по одній сторінці, висновки та пропозиції – 2 сторінки.

Заголовки структурних частин курсової роботи «ЗМІСТ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ ТА ПРОПОЗИЦІЇ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» (14 шрифт, жирно) пишуть великими літерами симетрично до тексту. Заголовки параграфів друкують маленькими літерами з абзацного відступу. Крапку в кінці заголовка не ставлять. Всі сторінки курсової роботи повинні бути пронумеровані арабськими цифрами без знака № в правому верхньому куті.

Першою сторінкою курсової роботи є титульний лист (приклад подано у **Додатку Б**), який включають до загальної нумерації сторінок. На ньому номер сторінки не ставлять.

На другій сторінці розміщують завдання на курсову роботу та календарний план її виконання (бланк завдання подано у **Додатку В**).

Змісті курсової роботи оформляється відповідно до **Додатку Г**. Зверху потрібно написати: **ЗМІСТ**, далі вказується кожна структурна частина та нумерація сторінок.

З наступної сторінки розпочинається виклад тексту роботи. Кожний розділ пишуть з нової сторінки і обов'язково вказують його найменування (виділяють заголовком). Номер розділу ставлять після слова «РОЗДІЛ», після номера крапку не ставлять, потім з нового рядка друкують заголовок розділу.

Параграфи нумерують у межах кожного розділу. Номер параграфу складається з номера розділу і порядкового номера параграфу, між якими ставлять крапку. В кінці номера параграфу повинна стояти крапка, наприклад: «2.1» (перший параграф другого розділу). Далі у тому ж рядку друкують заголовок параграфу.

Всі аркуші, на яких розміщені такі структурні частини курсової роботи як зміст, вступ, висновки, список використаних джерел, додатки нумерують звичайним чином. Не нумерують лише їх заголовки, тобто не можна друкувати «1. ВСТУП» або «4. ВИСНОВКИ».

Оформлення таблиць та ілюстрацій. Для аналізу необхідно використовувати таблиці та ілюстрації: схеми, графіки, діаграми тощо.

Цифровий матеріал, як правило, оформляють у вигляді таблиць. Кожна таблиця повинна мати назву, яку розміщують над таблицею симетрично до тексту. Назву і слово «Таблиця» починають з великої літери. Назву не підкреслюють. У дужках над змістом таблиці вказують одиниці виміру. Заголовки граф повинні починатися з великих літер, підзаголовки – з маленьких (якщо вони складають одне речення із заголовком) і з великих (якщо вони є самостійними).

Таблиці (**Додаток Д**) нумерують послідовно в межах розділу. В правому верхньому куті над відповідним заголовком таблиці розміщують напис «Таблиця» із зазначенням номера. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка, наприклад: «Таблиця 2.5» (п'ята таблиця другого розділу).

При переносі частини таблиці на інший аркуш слово «Таблиця» і її номер вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть «продовження таблиці» і вказують номер таблиці, наприклад: «Продовження таблиці 2.5». Таблицю з великою кількістю граф можна ділити на частини і розміщувати одну частину під іншою в межах однієї сторінки.

Ілюстрації позначають словом «Рис.» (**Додаток Е**) 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. Наприклад, **Рис. 1.1** (перший рисунок першого розділу). Номер ілюстрації, її назва і пояснювальні підписи розміщують послідовно під ілюстрацією. Якщо у роботі подано одну ілюстрацію, то її нумерують за загальними правилами.

Написання формул. При необхідності використовують формули. Їх виділяють з тексту вільними рядками. Над і під кожною формулою потрібно залишити не менше одного вільного рядка. Якщо рівняння чи формула не вміщаються в одному рядку, їх слід перенести після знака рівності (=) або після інших розділових знаків (+, -, :, ×). Невеликі і нескладні формули, що не мають самостійного значення, вписують всередині рядків тексту.

Пояснення значень символів і числових коефіцієнтів подають безпосередньо під формулою в тій послідовності, в якій вони наведені у формулі. Значення кожного символу і числового коефіцієнта треба подавати з нового рядка, який починають зі слова «де» без двокрапки. Нумерувати слід лише ті формули, на які є посилання у наступному тексті. Формули у курсовій роботі (якщо їх більше однієї) нумерують у межах розділу. Номер формули складається з номера розділу і порядкового номера формули в розділі, між якими ставлять крапку. Нумери формул пишуть біля правого поля аркуша на рівні відповідної формули в круглих дужках, наприклад: (2.3) (третя формула другого розділу) (**Додаток Ж**).

Посилання на використані джерела. При написанні курсової роботи студент повинен робити посилання на використані літературні джерела. Посилання робляться тоді, коли в тексті використовують цитати чи вислови або переповідається своїми словами думка того чи іншого автора, наводяться цифри і факти, використовуються таблиці або ілюстрації з певного джерела тощо. Вони дають необхідну інформацію щодо процитованого матеріалу, допомагають з'ясувати його зміст, мову тексту, а також дають змогу відшукати документи і перевірити їх достовірність.

Загальні вимоги до цитування:

- текст цитати береться в лапки і наводиться дослівно;
- цитування повинно бути повним (у випадку пропуску слів в цитаті ставляться три крапки), без перекручень думок автора;
- кожна цитата повинна мати посилання на джерело;

▪ при непрямому цитуванні слід також посилатись на джерело, а думка автора повинна передаватись максимально точно.

Посилання в тексті курсової роботи роблять згідно з їх переліком у квадратних дужках, наприклад [2, с. 25], де перша цифра – номер джерела у списку використаних джерел, друга – сторінка, з якої взята цитата. Якщо посилання робиться лише на джерело (без використання цитати), – у квадратних дужках вказується тільки одна цифра, яка відображає порядковий номер джерела у списку використаних джерел, наприклад [25]. Посилатися слід на останні видання публікації.

У повторних посиланнях на таблиці та ілюстрації пишуть скорочено слово «дивись», наприклад: «(табл. 3.2)». При цьому слово «таблиця» в тексті пишуть скорочено. Посилаючись на джерело, слід мати на увазі, що кількість посилань має бути оптимальною, тобто не надто надмірною чи мізерною.

Формування списку використаних джерел. Список використаних джерел розміщують після висновків. У списку повинні наводитись лише ті джерела, які дійсно використовувалися автором при написанні курсової роботи.

Список використаних джерел формується в алфавітному порядку прізвищ перших авторів або заголовків.

Слід наголосити, що назва джерел подається тією мовою, якою вони видані. Не потрібно перекладати їх на українську.

Забороняється використання джерел, які опубліковані мовою країни-агресора.

Приклад оформлення списку використаних джерел наведено в **Додатку 3**. Бібліографічний опис джерел складають відповідно до чинних стандартів з бібліотечної та видавничої справи, міжнародних і державного стандартів з обов'язковим наведенням назв праць.

Оформлення додатків. Додатки оформляються після списку використаних джерел з нової сторінки з назвою «ДОДАТКИ». Додатки потрібно подавати за таким зразком:

ДОДАТОК А

Зразок оформлення титульної сторінки курсової роботи

.....

Текст / таблиця / малюнок додатку

2. ЗАХИСТ ТА КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ

Захист курсових робіт відбувається у дні та години, вказані у графіку захисту курсових робіт або відповідно до розкладу занять. Під час захисту студент коротко висловлює зміст роботи, відповідає на запитання викладача та зауваження, висловлені у рецензії. Оцінка за курсову роботу залежить від повноти викладу теми, правильності висвітлення питань, рівня використання теоретичного, методологічного та фактологічного матеріалу; належного оформлення та рівня захисту роботи. Підсумкову, диференційовану оцінку курсової роботи визначає комісія, що призначається наказом керівника закладу освіти.

Курсова робота не допускається до захисту і повертається на доопрацювання, якщо:

- роботу представлено циклової комісії (на перевірку) для рецензування з порушенням термінів, установлених цикловою комісією (викладачем, який викладає дану дисципліну);
- роботу написано на тему, що не включена до переліку тем курсових робіт з даної дисципліни або не погоджена з викладачем;
- роботу виконано не самостійно;
- структура і логіка побудови плану роботи не відповідає вимогам та темі курсової роботи;

У процесі оцінювання враховується низка важливих показників якості курсової роботи, а саме:

- чіткість формулювання мети і завдання курсової роботи, складність досліджуваних у роботі проблем, відповідність логічної побудови роботи поставленим цілям і завданням;
- якість і глибина теоретико-методичного аналізу проблеми; широта й адекватність методологічного апарату; якість критичного огляду літературних джерел, наявність наукової полеміки, посилань на літературні джерела та визначення власної думки студента-автора курсової роботи;
- системність і глибина аналізу статистичних та фактичних матеріалів, наявність і переконливість узагальнень і висновків з аналізу, наявність і якість ілюстративних матеріалів у тексті роботи, використання економіко-математичних методів, соціологічних та інших досліджень;
- наявність та логічний зв'язок заходів, що пропонуються для вирішення проблеми, що досліджується у роботі, з проведеним у роботі аналізом фактичних та статистичних матеріалів, їх актуальність, реальність та економіко-соціальна обґрунтованість, наявність альтернативних підходів до вирішення визначених проблем;
- володіння культурою презентації, вміння стисло (в межах регламенту), послідовно й чітко викласти сутність і результати дослідження, здатність аргументовано захищати свої пропозиції, думки, погляди; повнота і ґрунтовність відповідей на запитання членів комісії, що приймають захист курсових робіт, на зауваження і пропозиції, що містяться у рецензії на курсову роботу.

3. ДОТРИМАННЯ НОРМ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Студенти при виконанні курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» повинні дотримуватись норм академічної доброчесності, тобто самостійно виконувати курсову роботу, здійснювати посилання на джерела інформації у разі використання певних тверджень й відомостей, надавати достовірну інформацію про результати власних розробок та досліджень. У курсовій роботі

доцільно уникати плагіату, фальсифікації та фабрикування матеріалів дослідження.

Академічний плагіат – це оприлюднення (частково або повністю) наукових результатів, отриманих іншими особами, як результатів власного дослідження та/або відтворення опублікованих текстів інших авторів без зазначення авторства.

Академічним плагіатом є:

1) відтворення в тексті роботи без змін, з незначними змінами, або в перекладі тексту іншого автора (інших авторів), обсягом від речення і більше, без посилання на автора (авторів) відтвореного тексту;

2) відтворення в тексті роботи повністю або частково, тексту іншого автора (інших авторів) через його перефразування чи довільний переказ без посилання на автора (авторів) відтвореного тексту;

3) відтворення в тексті роботи наведених в іншому джерелі цитат з третіх джерел без вказування, за яким саме безпосереднім джерелом наведена цитата;

4) відтворення в тексті роботи наведеної в іншому джерелі науково-технічної інформації (крім загальновідомої) без вказування на те, з якого джерела взята ця інформація.

Фальсифікація розуміється, як свідомо зміна чи модифікація вже наявних даних, зокрема підробка підписів в офіційних документах.

Під **фабрикуванням** мається на увазі вигадкування даних чи фактів, що використовуються у власній діяльності чи діяльності інших учасників в процесі реалізації державної політики в сфері якості освіти.

Перевірка на плагіат здійснюється відповідно до *«Положення про систему запобігання та виявлення академічного плагіату у Фаховому коледжі ЗВО «Університет Короля Данила»*.

Курсова робота, яка виконана не самостійно, з грубими помилками, в якій відсутні посилання на використані літературні джерела, містить ознаки плагіату чи інші грубі помилки повертається автору.

У такому випадку повторно виконується вся робота, з урахуванням вказівок, зазначених у рецензії і разом із нею знову повертається до Фахового коледжу.

До курсової роботи додається довідка довільного взірця про відсоток оригінального тексту або відсоток плагіату.

4. ТЕМИ КУРСОВИХ РОБІТ

Основна частина курсової роботи має містити виконання наступних завдань:

- створити класовий тип згідно із варіантом;
- для створення об'єкту динамічного типу передбачити відповідні конструктори (конструктор по замовчанню, конструктор із параметрами, конструктор копіювання);
- реалізувати компонентні методи відповідно до варіанту;
- розробити демонстраційно-тестувальну програму; виконати тестування розробленого програмного коду.

1. Класовий тип для обробки текстової інформації.

2. Класові типи для роботи з чисельними матрицями та векторами.

3. Бібліотечні засоби для розв'язування задач лінійної алгебри.

4. Класовий тип для роботи зі структурами типу «Вектор».

5. Класовий тип для роботи зі структурами типу «Бітовий вектор».

6. Клас для роботи зі структурами типу «Зв'язний список».

7. Клас для обслуговування структур типу «Черга».

8. Клас для обслуговування структур типу «Двобічна черга».
9. Клас для обслуговування структур типу «Черга з пріоритетами».
10. Клас для роботи з структурами типу «Множина».
11. Клас для роботи з структурами типу «Мультимножина».
12. Клас для роботи з структурами типу «Словник».
13. Клас для роботи з структурами типу «Словник з повтореннями».
14. Клас для роботи з структурами типу «Стек».
15. Клас для роботи з структурами типу «Дерево».
16. Клас для роботи з структурами типу «Врівноважене дерево».
17. Класи для роботи з неорієнтованими графами: таблиця зв'язків та матриця суміжності.
18. Класи для роботи з орієнтованими графами: таблиця зв'язків та матриця суміжності.
19. Класи для компактного подання неорієнтованих та орієнтованих графів.
20. Клас для роботи з дводольними графами.
21. Клас для роботи з мультиграфами.
22. Клас для роботи з гіперграфами.
23. Клас для роботи з табличними інваріантами графів.
24. Розробка та реалізація алгоритму упізнання ізоморфності графів.
25. Модель лінії затримки на основі структури типу «Черга».
26. Комп'ютерна модель системи терморегулювання.
27. Алгебра бінарних відношень. Програмна реалізація.

ВАРІАНТИ ЗАВДАНЬ ДО КУРСОВОЇ РОБОТИ

Варіант 1. Класовий тип для обробки текстової інформації.

1. Створити класовий тип String - символний масив динамічного типу із змінною кількістю символів. Максимальний розмір масиву - у межах вільного простору оперативної пам'яті.
2. Реалізувати компонентні методи:
 - довжина рядка;
 - пошук субрядку;
 - пошук субрядку та його заміна;
 - приведення усіх букв до верхнього регістру;
 - приведення усіх букв до нижнього регістру;
 - конкатенація;
 - обмін значеннями двох змінних типу String.
 Література: [1-5, 7].

Варіант 2. Класові типи для роботи з чисельними матрицями та векторами

1. Створити класовий тип Matrix - двовимірний чисельний масив динамічного типу із змінними розмірами.
2. Створити класовий тип Vector - одновимірний чисельний масив динамічного типу із змінними розмірами.
3. Для конкретизації типів матриці та вектора використати шаблони.
4. Реалізувати компонентні методи:
 - транспонування матриці;
 - множення матриць;
 - складання та віднімання;

- порівняння матриць;
- множення матриці на вектор;
- скалярний добуток векторів;
- обчислення норми матриці;
- пошук седлових точок матриці;
- пошук мінімальних та максимальних елементів.

Література: [1-5, 7, 14].

Варіант 3. Бібліотечні засоби для розв'язування задач лінійної алгебри

1. Створити класові типи - чисельна квадратна матриця та одновимірний масив динамічного типу із змінними розмірами.

2. Створити перевантажені операції:

3. Реалізувати компонентні методи:

- додавання та віднімання матриць;
- додавання та віднімання векторів;
- множення матриці на вектор;
- обертання матриці;
- порівняння матриць;
- розв'язування системи лінійних рівнянь;
- скалярне множення векторів.
- обчислення визначника матриці;
- обчислення довжини вектору.

Література: [1-5, 7, 14].

Варіант 4. Класовий тип для роботи з структурами типу «Вектор»

1. Створити параметризований клас для роботи з структурою типу Vector - одновимірний динамічний масив із змінною кількістю елементів. Тип елементу масиву надається параметром шаблону.

2. Передбачити конструктор для створення вектору та ініціалізації його звичайним одновимірним масивом відповідного типу.

3. Реалізувати компонентні методи:

- get - доступу до елемента за індексом;
- concat - конкатенації двох векторів;
- equals - порівняння двох векторів.
- size - кількість елементів вектора;
- front - посилання на перший елемент;
- back - посилання на останній елемент;
- swap - обмін значеннями з іншим вектором;
- insert - уставити елемент у надану позицію;
- push_back - додати новий елемент у кінець вектора;
- pop_back - вилучити останній елемент;
- erase - вилучити елемент у наданій позиції;
- find - знайти елемент у векторі і повернути його позицію;
- accumulate - накопичення суми або добутку;
- for_each - обробка елементів по наданій процедурі;
- max, min - пошук максимального та мінімального елементів;
- sort - упорядкування елементів у порядку зростання або зменшення.

Література: [1-6].

Варіант 5. Класовий тип для роботи з структурами типу «Бітовий вектор»

1. Створити параметризований клас для роботи з структурою типу BitVector, яка являє собою одновимірний динамічний масив із змінною кількістю елементів бітового типу.

2. Реалізувати компонентні методи:

- get - доступу до бітового елемента за індексом;
- concat - конкатенації двох бітових векторів;
- equals - порівняння;
- бітові операції AND, OR, XOR, NOT.
- push_back - додання нового бітового елемента у кінець вектора;
- pop_back - вилучення останнього бітового елемента;
- size - кількість елементів вектора;
- front - посилання на перший елемент;
- back - посилання на останній елемент;
- swap - обмін значеннями з іншим вектором;
- num_true - кількість елементів, що дорівнюють true;
- num_false - кількість елементів, що дорівнюють false.

Література: [1-6].

Варіант 6. Клас для роботи з структурами типу «Зв'язний список»

1. Створити клас List для роботи із структурами типу «Двонаправлений зв'язний список». Тип елемента списку надається параметром шаблону. Елемент двонапрявленого зв'язного списку містить у собі покажчики на наступний та попередній елементи структури last і next. Значення last для першого елемента і значення next для останнього дорівнюють NULL.

2. Реалізувати компонентні методи:

- size - кількість елементів списку;
- front - посилання на перший елемент;
- back - посилання на останній елемент;
- insert - уставити елемент у надану позицію;
- push_back - додати новий елемент у кінець списку;
- pop_back - вилучити останній елемент;
- push_front - додати новий елемент у початок списку;
- pop_front - вилучити останній елемент;
- find - перевірити, чи є наданий елемент у списку;
- remove - вилучення всіх елементів із списку з наданим значенням;
- accumulate - накопичення суми або добутку;
- for_each - обробка елементів по наданій процедурі;
- max, min - пошук максимального та мінімального елементів;
- sort - упорядкування елементів по зростанню;
- unique - вилучення повторень (якщо два сусідніх елементи співпадають, один з них вилучається);
- filter - утворити новий список з елементів, що відповідають заданій умові (пердикату);
- split - розбиття вихідного списку на два, залежно від умови (предикату);

- swap - обмін значеннями з іншим списком;
- equals - поелементне порівняння двох списків;
- merge - об'єднання (конкатенація) з наданим списком.

Література: [1-5, 9].

Варіант 7. Клас для обслуговування структур типу «Черга»

1. Створити клас Queue для роботи із структурами типу «Черга». Тип елементу черги надається параметром шаблону. Черга є структурою даних типу «Перший увійшов - перший вийшов» (FIFO - first-in, first-out), вона нагадує ескалатор у метро.

2. Реалізувати компонентні методи:

- equals - порівняння;
- get - доступу по індексу;
- size - кількість елементів;
- empty - повертає значення true, якщо черга пуста;
- front - посилання на перший елемент;
- back - посилання на останній елемент;
- swap - обмін значеннями з іншою чергою;
- push - додати новий елемент у кінець черги;
- pop - вилучити перший елемент;
- find - перевірити, чи є наданий елемент у черзі;
- accumulate - накопичення суми або добутку;
- for_each - обробка елементів по наданій процедурі;
- max , min - пошук максимального та мінімального елементів.

Література: [1-5, 9].

Варіант 8. Клас для обслуговування структур типу «Дек»

1. Створити клас Deque для роботи із структурами типу «Дек» (Двобічна черга). Тип елементу дека надається параметром шаблону.

Дек є черга з двостороннім доступом (deque - «double ended queue»). Деки використовуються з алгоритмами, які потребують доступу до голови і хвоста структури. Він поєднує у собі можливості вектора і списку. Для нього визначається операція індексування. Додавати і вилучати елементи дека можна з обох сторін.

2. Реалізувати компонентні методи:

- порівняння;
- доступу по індексу;
- size - кількість елементів;
- empty - повертає значення true, якщо дек пустий;
- front - посилання на перший елемент;
- back - посилання на останній елемент;
- swap - обмін значеннями з іншим деком;
- push_front - додати новий елемент у початок дека;
- push_back - додати новий елемент у кінець дека;
- pop_front - вилучити перший елемент;
- pop_back - вилучити останній елемент;
- find - перевірити, чи є наданий елемент у деку;
- accumulate - накопичення суми або добутку;
- for_each - обробка елементів по наданій процедурі;

- max , min - пошук максимального та мінімального елементів.

Література: [1-5, 9].

Варіант 9. Клас для обслуговування структур типу «Черга з пріоритетами»

1. Створити клас PriorityQueue для роботи із структурами типу «Черга з пріоритетами». Тип елемента структури надається параметром шаблону.

Структура типу «Черга з пріоритетами» має такі ж властивості, як і проста черга, але елементи її завжди розташовані у певному порядку. Упорядкування структури виконується автоматично при додаванні нового елемента. Тому операція pop вилучає найбільший (чи найменший) елемент черги.

2. Реалізувати компонентні методи:

- equals - порівняння;
- size - кількість елементів;
- empty - повертає значення true, якщо черга пуста;
- back - посилання на останній елемент;
- swap - обмін значеннями з іншою чергою;
- push - додати новий елемент у кінець черги;
- pop - вилучити найбільший (найменший) елемент;
- find - перевірити, чи є наданий елемент у деку;
- accumulate - накопичення суми або добутку;
- for_each - обробка елементів по наданій процедурі.

Література: [1-5, 9].

Варіант 10. Клас для роботи з структурами типу «Множина»

1. Створити клас Set для роботи із структурами типу «Множина». Кожен елемент множини має унікальне значення. Тип елемента структури надається параметром шаблону.

Множина є набір елементів, кожен з яких має унікальне значення. При додаванні елемента (операція include), який вже є у множині, він не додається. Елементи множини називають ключами. Для цієї структури визначаються операції, звичайні для математичних множин - об'єднання, переріз, віднімання, доповнення. Елементи множини автоматично упорядковуються, це дає змогу використовувати швидкі операції пошуку елемента із наданим значенням.

2. Реалізувати компонентні методи:

- equals - порівняння;
- union - об'єднання множин;
- section - переріз множин;
- subtract - віднімання;
- size - кількість елементів;
- empty - повертає значення true, якщо множина пуста;
- swap - обмін значеннями з іншою множиною;
- include - додати новий елемент у множину;
- exclude - вилучити наданий елемент із множини;
- compl - отримати значення, яке є доповненням до наданої множини;
- find - перевірити, чи є наданий елемент у множині;
- accumulate - накопичення суми або добутку для всіх елементів множини;
- for_each - змінювання елементів згідно з наданою процедурою.

Література: [1-5, 9].

Варіант 11. Клас для роботи з структурами типу «Мультимножина»

1. Створити клас MultiSet для роботи із структурами типу «Мультимножина». Тип елементу структури надається параметром шаблону.

Ця структура має такі ж властивості, як і множина, але елемент мультимножини не обов'язково має унікальне значення (елементи можуть повторюватись).

2. Реалізувати компонентні методи:

- equals - порівняння;
- size - кількість елементів;
- empty - повертає значення true, якщо мультимножина пуста;
- swap - обмін значеннями з іншою структурою того ж типу;
- include - додати новий елемент у мультимножину;
- exclude - вилучити наданий елемент із мультимножини;
- find - перевірити, чи є наданий елемент у мультимножині;
- accumulate - накопичення суми або добутку для всіх елементів структури;
- for_each - змінювання елементів згідно з наданою процедурою.

Література: [1-5, 9].

Варіант 12. Клас для роботи з структурами типу «Словник»

1. Створити клас Map для роботи із структурами типу «Словник» (таку структуру називають ще асоціативний масив або відображення). Елементом цієї структури є пара: ключ і значення. Значення ключа є унікальним. Словник автоматично упорядковується по ключам. Типи компонентів пари - елемента словника надається параметрами шаблону. Для словника звичайно визначаються операція індексування для швидкого доступу до ключа і операція порівняння для ключів.

Застосовуються такі структури у задачах із словниками і асоціативними базами даних.

2. Реалізувати компонентні методи:

- equals - порівняння;
- size - кількість елементів;
- size - кількість елементів;
- empty - повертає значення true, якщо словник пустий;
- swap - обмін значеннями з іншим словником;
- include - додати елемент з наданим ключем у словник;
- exclude - вилучити елемент з наданим ключем із словника;
- find - знайти значення, яке відповідає наданому ключу;
- for_each - змінювання елементів згідно з наданою процедурою.

Література: [1-5, 9].

Варіант 13. Клас для роботи з структурами типу «Словник з повтореннями»

1. Створити клас MultiMap для роботи із структурами типу «Словник з повтореннями» (мультивідображення). Типи компонентів пари - елемента словника надається параметрами шаблону.

Ця структура має такі ж властивості, як і попередня (словник), але ключі елементів мультивідображення не обов'язково мають унікальне значення, структура може вміщувати елементи з однаковими значеннями ключів. Мультивідображення ,

також як і словник, автоматично упорядковується по ключам.

2. Реалізувати компонентні методи:

- equals - порівняння;
- size - кількість елементів;
- empty - повертає значення true, якщо словник порожній;
- swap - обмін значеннями з іншим словником;
- include - додати елемент з наданим ключем у словник;
- erase - вилучити всі елементи з наданим ключем із словника і повернути кількість вилучених елементів;
- find - знайти наступне значення, яке відповідає наданому ключу;
- for_each - змінювання елементів згідно з наданою процедурою.

Література: [1-5, 9].

Варіант 14. Клас для роботи з структурами типу «Стек»

1. Створити клас для роботи із структурами типу «Стек» (структура типу «останній ввійшов - перший вийшов», або LIFO - last-in, first-out). Тип компонентів структури надається параметрами шаблону.

Структура типу «Стек» може створюватись на основі однієї з таких структур: вектор, дек або список. В усіх випадках для елементів стека визначаються однаковий набір операцій. Стек повинен піддержувати операції порівняння.

Структури цього типу застосовуються у багатьох алгоритмах, особливо для збереження і здобуття інформації у певному порядку (наприклад, для результатів обчислень підвиразів у електронному калькуляторі).

2. Реалізувати компонентні методи:

- equals - порівняння;
- size - кількість елементів у поточний момент;
- empty - повертає значення true, якщо стек пустий;
- swap - обмін значеннями з іншим стеком;
- push - ввести значення у стек;
- pop - вилучити значення з вершини стека;
- top - доступ до вершини стека.

Література: [1-5, 9].

Варіант 15. Клас для роботи з структурами типу «Дерево»

1. Створити клас Tree для роботи із структурами типу «Дерево». Тип елемента структури надається параметром шаблону. Розробити метод адресації елемента дерева.

2. Для створення об'єкту динамічного типу і правильного його вилучення передбачити відповідні конструктори та деструктори. Для ініціалізації об'єктів передбачити конструктор копіювання та конструктори з параметрами.

2. Реалізувати компонентні методи:

- equals - порівняння;
- pop - отримати наступний елемент дерева (у порядку його послідовного обходу у глибину);
- numbel - загальна кількість вузлів дерева;
- numbl - кількість кінцевих вузлів дерева;
- height - висота дерева;
- empty - повертає значення true, якщо дерево пусте;
- swap - обмін значеннями двох змінних типу Tree;

- include - додати новий елемент у дерево;
- exclude - вилучити наданий елемент із дерева;
- find - перевірити, чи є наданий елемент у дереві;
- accumulate - накопичення суми або добутку для всіх елементів дерева;
- for_each - змінювання елементів згідно з наданою процедурою.

Література: [1-5, 9].

Варіант 16. Клас для роботи з структурами типу «Врівноважене дерево»

1. Створити клас AVLTree для роботи із структурами типу «Врівноважене дерево». Тип елемента структури надається параметром шаблону. Розробити метод адресації елемента дерева. Передбачити, щоб після виконання операції додавання нового вузла, дерево автоматично врівноважувалось.

2. Реалізувати компонентні методи:

- equals - порівняння;
- pop - отримати наступний елемент дерева (у порядку його послідовного обходу у глибину);
- numbel - загальна кількість вузлів дерева;
- numbl - кількість кінцевих вузлів дерева;
- height - висота дерева;
- empty - повертає значення true, якщо дерево пuste;
- swap - обмін значеннями двох змінних типу Tree;
- include - додати новий елемент у дерево;
- exclude - вилучити наданий елемент із дерева;
- find - перевірити, чи є наданий елемент у дереві;
- accumulate - накопичення суми або добутку для всіх елементів дерева.

Література: [1-5, 9].

Варіант 17. Класи для роботи з неорієнтованими графами: таблиця зв'язків та матриця суміжності

1. Створити класові типи - таблиця зв'язків та матриця суміжності неорієнтованого графа.

2. Для створення об'єктів динамічного типу і правильного їх знищення передбачити відповідні конструктори та деструктори. Для ініціалізації об'єктів передбачити конструктори копіювання та відповідні конструктори з параметрами і конструктори перетворення.

2. Реалізувати компонентні методи:

- вилучення даної вершини із графа;
- додання вершини у граф;
- вилучення даного ребра із графа;
- додання даного ребра у граф;
- функцію, що повертає впорядкований вектор степенів вершин.

Література: [1-5, 7, 10-12].

Варіант 18. Класи для роботи з орієнтованими графами: таблиця зв'язків та матриця суміжності

1. Створити класові типи - таблиця зв'язків та матриця суміжності орієнтованого графа.

2. Реалізувати компонентні методи:

- процедуру вилучення даної вершини із графа;

- процедуру добавлення вершини у граф;
- процедуру вилучення даного ребра із графа;
- процедуру добавлення даного ребра у граф;
- функцію, що повертає впорядкований вектор степенів вершин.
- Література: [1-5, 7, 10-12].

Варіант 19. Класи для компактного подання неорієнтованих та орієнтованих графів

1. Створити класові типи, що відображають бінарні коди неорієнтованого та орієнтованого графа.

2. Реалізувати компонентні методи:

- процедуру вилучення даної вершини із графа;
- процедуру додавання вершини у граф;
- процедуру вилучення даного ребра із графа;
- процедуру добавлення даного ребра у граф;
- функцію, що повертає впорядкований вектор степенів вершин;
- процедуру, що формує таку множину графів, що утворюється при додаванні одного ребра у даний граф.

Література: [1-5, 7, 10-12].

Варіант 20. Клас для роботи з дводольними графами

1. Розробити класовий тип для подання та виконання операцій над дводольними графами.

2. Розробити алгоритм упізнання дводольності графа. Застосувати метод, що базується на хвильовому процесі.

3. Реалізувати компонентні методи:

- процедуру вилучення даної вершини із графа;
- процедуру додавання вершини у граф;
- процедуру вилучення даного ребра із графа;
- процедуру добавлення даного ребра у граф;
- функцію, що повертає впорядковані вектори степенів вершин першої та другої долі.

Література: [1-5, 7, 10-12].

Варіант 21. Клас для роботи з мультіграфами

1. Створити класовий тип для виконання операцій з мультіграфами.

2. Реалізувати компонентні методи:

- процедуру вилучення даної вершини із графа;
- процедуру добавлення вершини у граф;
- процедуру вилучення даного ребра із графа;
- процедуру добавлення даного ребра у граф;
- функцію, що повертає впорядкований вектор степенів вершин.

Література: [1-5, 10-12].

Варіант 22. Клас для роботи з гіперграфами

1. Створити класовий тип для виконання операцій з гіперграфами.

Застосувати уявлення гіперграфу як напрямлений дводольний граф.

2. Реалізувати компонентні методи:

- процедуру вилучення даної вершини із графа;
- процедуру добавлення вершини у граф;

- процедуру вилучення даного ребра із графа;
- процедуру добавлення даного ребра у граф;
- функцію, що повертає впорядкований вектор степенів вершин;
- функцію, що повертає впорядкований вектор потужностей ребер.

Література: [1-5, 10-12].

Варіант 23. Клас для роботи з табличними інваріантами графів

1. Створити класовий тип для табличних інваріантів графа першого порядку.
2. Реалізувати компонентні методи:
 - функцію, що повертає скалярне інтегроване значення інваріанту;
 - функцію, що повертає впорядкований вектор степенів вершин;
 - функції, що повертають діаметр та радіус графа;
 - функцію, що упізнає дводольність графа.

Література: [1-5, 10-12].

Варіант 24. Розробка та реалізація алгоритму упізнання ізоморфності графів

1. Створити алгоритм та програму встановлення ізоморфності двох даних графів переборним методом.
2. Побудувати демонстраційно-тестуючу програму, що розв'язує такі задачі:
 - задачу класифікації для даної множини графів;
 - задачу класифікації для вершин даного графа;
 - задачу класифікації для ребер даного графа.
3. Виконати дослідження обчислювальної ефективності створеного алгоритму упізнання ізоморфності графів.

Література: [1-5, 10-12].

Варіант 25. Модель лінії затримки на основі структури типу «Черга»

1. Створити комп'ютерну модель узагальненої лінії затримки на основі структури типу «Черга», застосовуючи Java Collections Framework.
 2. Модель повинна відображати такі ефекти:
 - затримку сигналу, що пов'язана з скінченою швидкістю розповсюдження сигналу;
 - зниження величини сигналу завдяки поглинанню енергії у матеріалі лінії задержки;
 - модифікацію форми сигналу як результат дії ефекту дисперсії.
 3. Передбачити у програмі графічне виведення для сигналів, що діють на вході і на виході лінії затримки.
 4. Виконати розрахунки для сигналів прямокутної та трикутної форми.
- Література: [1-5].

Варіант 26. Побудова засобів обробки даних на основі Java Collections Framework

1. Розробити бібліотеку процедур та функцій для побудови програм, де застосовуються операції обробки чисельних даних методом найменших квадратів на основі Java Collections Framework.
2. Передбачити процедури та функції для розв'язування таких задач:
 - апроксимація даних лінійною функцією;
 - апроксимація даних поліномом даної степені;
 - апроксимація даних кубічними сплайнами.

3. Для побудови процедур створити і застосувати класові типи «Динамічна матриця» і «Динамічний вектор».

Література: [1-5, 6].

Варіант 27. Побудова засобів для обчислень з довільною точністю на основі Java Collections Framework

1. Створити класовий тип `Large` для подання значення дійсного типу з наданим числом двійкових розрядів мантиси. Застосувати засоби стандартної бібліотеки `Java Collections Framework`.

2. Реалізувати компонентні методи:

- присвоєння;
- додавання;
- віднімання;
- множення;
- ділення;

3. На основі створеного класу написати програму обчислення константи π з наданим числом десяткових розрядів.

Література: [1-6].

Варіант 28. Комп'ютерна модель системи терморегулювання

1. Розробити засоби комп'ютерного моделювання систем автоматичного регулювання температурою об'єктів. Створити класи:

- об'єкт;
- зовнішнє середовище;
- регулятор;
- функція цілі.

Передбачити можливість регулювання за законами пропорційного, інтегрального та диференціального регулювання.

2. Створити компонентні функції (процедури) для кожного класу для отримання нового стану відповідного об'єкту у наступний момент часу. Для передачі повідомлень від одного об'єкту до другого застосувати дружні функції.

3. Усі класи створити з урахуванням правил об'єктно-орієнтованого програмування.

4. На базі створених класів написати програму моделювання та дослідження систем терморегулювання з різними законами регулювання.

5. Дослідити можливості створених засобів, сформулювати задачі проектування, які можуть бути розв'язані за допомогою створених засобів.

Література: [1-7].

Варіант 29. Алгебра бінарних відношень. Програмна реалізація

1. Створити класовий тип `binrel` для роботи з бінарними відношеннями.

2. Для створення об'єкту динамічного типу і правильного його вилучення передбачити відповідні конструктори та деструктори. Для ініціалізації об'єктів передбачити конструктор копіювання.

2. Реалізувати компонентні методи:

- знаходження зворотнього відношення по наданому відношенню;
- отримання композиції двох відношень.
- функцію, яка визначає, чи є надане відношення рефлексивним;
- функцію, яка визначає, чи є надане відношення симетричним;
- функцію, яка визначає, чи є надане відношення антисиметричним;
- функцію, яка визначає, чи є надане відношення транзитивним;

- функцію, яка визначає, чи є надане відношення функцією;
- функцію, яка визначає, чи є надане відношення ін'єктивною функцією;
- функцію, яка визначає, чи є надане відношення сюр'єктивною функцією;
- функцію, яка визначає, чи є надане відношення бієктивною функцією;
- функцію, яка визначає, чи є надане відношення відношенням частинного порядку;
- функцію, яка визначає, чи є надане відношення відношенням лінійного порядку;
- функцію, яка визначає, чи є надане відношення відношенням еквівалентності.

Література: [1-5, 15].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bloch J. Effective Java: Programming Language Guide. Publisher: Addison Wesley, 2001. – 180 p.
2. The Java Tutorials [Electronic resource]. URL: <http://download.oracle.com/javase/tutorial> (дата звернення: 20.07.2022 р.).
3. Programming Tutorials and Source Code Examples. URL: <http://www.java2s.com> (дата звернення: 20.07.2022 р.).
4. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник]. К.: ІТкнига, 2015. 624 с. URL: <https://cutt.ly/XwqrpqLL> (дата звернення: 20.07.2022 р.).
5. Настенко Д. В., Нестерко А Б. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові С+.: Навчальний посібник. К.: НТУУ «КПІ», 2016. 76 с. URL: https://ela.kpi.ua/bitstream/123456789/16671/1/OOP_manual.pdf (дата звернення: 20.07.2022 р.).
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://cutt.ly/FwqrpWVl> (дата звернення: 20.07.2022 р.).

ДОДАТКИ**ДОДАТОК А****Форма заяви студента на виконання курсової роботи**

Голові циклової комісії
з інформаційних технологій
Наталії МАЛИНОВСЬКІЙ

.....
.....
.....
(Ім'я ПРІЗВИЩЕ здобувача освіти,
шифр групи, спеціальність, номер
телефону)

ЗАЯВА

Прошу дозволити написання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» на тему

.....
.....

(вказати тему курсової роботи)

під керівництвом

(вказати прізвище, ім'я, по батькові наукового керівника курсової роботи)

.....
(дата)

.....
(підпис)

Взірець титульного аркуша курсової роботи

**ФАХОВИЙ КОЛЕДЖ
ЗВО «УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

Циклова комісія з інформаційних технологій

КУРСОВА РОБОТА
з дисципліни «Об'єктно-орієнтоване програмування» на тему
«КЛАСОВИЙ ТИП ДЛЯ ОБРОБКИ ТЕКСТОВОЇ ІНФОРМАЦІЇ»

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**

студента (-ки) 3 курсу
галузі знань 12 Інформаційні технології,
спеціальності 121 Інженерія програмного
забезпечення

.....
(*прізвище та ініціали*)

Науковий керівник:
к.т.н. Пашкевич О.П.

Оцінка за шкалою:

національною _____

100-бальною _____

за шкалою ECTS _____

Члени комісії:

.....
(*підпис*) (*прізвище та ініціали*)

.....
(*підпис*) (*прізвище та ініціали*)

Бланк завдання на курсовий проект (роботу)

_____ назва закладу освіти
 Циклова комісія _____
 Дисципліна _____
 Спеціальність _____
 Курс _____ Група _____ Семестр _____

ЗАВДАННЯ на курсовий проект (роботу) студента

- _____ Прізвище, ім'я, по-батькові
1. Тема проекту (роботи) _____
 2. Строк здачі студентом закінченого проекту (роботи) _____
 3. Вихідні дані до проекту (роботи) _____
 4. Зміст роботи (перелік питань по розділах, які потрібно розробити) _____
 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
 6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсового проекту (роботи)	Строк виконання	Примітки

Студент _____

Підпис

Ім'я, ПРИЗВИЩЕ

Керівник _____

Підпис

Ім'я, ПРИЗВИЩЕ

« ____ » _____ 202__ р.

Приклад оформлення змісту курсової роботи

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Аналіз стану технологій програмування та обґрунтування теми	6
РОЗДІЛ 2. Розробка програми виконання завдання	12
2.1. Розробка методу вирішення задачі.....	12
2.2. Структура даних і функцій.....	14
РОЗДІЛ 3. Розробка програми меню	20
3.1. Розробка та виконання тестового прикладу.....	21
3.2. Інструкція користувача.....	22
ВИСНОВОК	24
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	26
ПРАКТИЧНА ЧАСТИНА	30

Приклад оформлення таблиць у курсовій роботі

Таблиця 2.1 – Показники отримані підчас першого експерименту у несприятливих для GPS навігатора умовах

№	довгота	широта	№	довгота	широта
1	24.9218540	48.9218540	12	24.7003985	48.9218805
2	24.7003580	48.9218699	13	24.7003982	48.9218851
3	24.7003644	48.9218615	14	24.7003961	48.9218894
4	24.7003615	48.9218710	15	24.7003970	48.9218939
5	24.7003694	48.9218783	16	24.7004020	48.9218970
6	24.7003807	48.9218781	17	24.7004060	48.9219006

Приклад оформлення рисунків у курсовій роботі

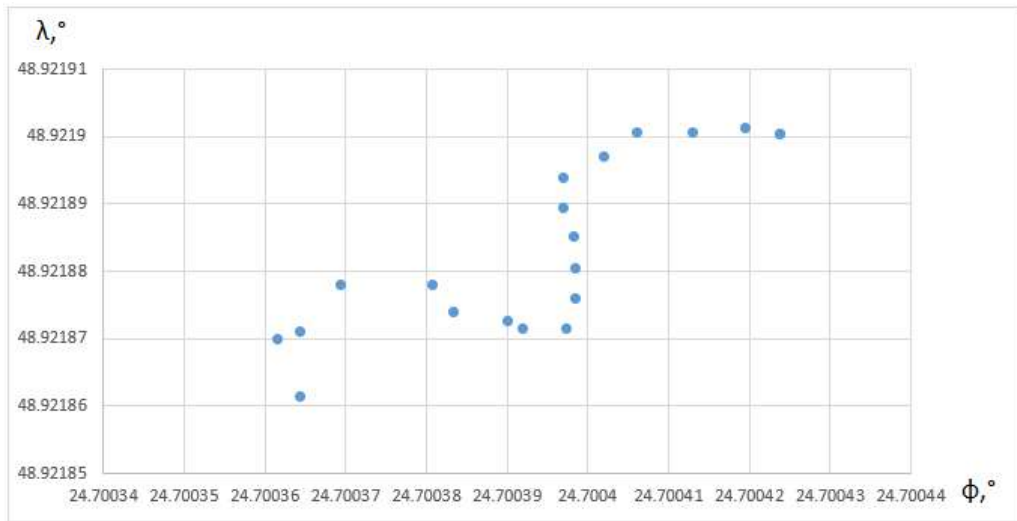


Рисунок 2.7 – Діаграма множин значень отриманих від GPS модуля на одній позиції у несприятливих умовах

Взірець подання формул у курсовій роботі

Коефіцієнти передавальної функції цифрового фільтру нижніх частот першого порядку повинні відповідати формулі 2.2

$$b_0 = b_1 = (1 + a)/2. \quad (2.2)$$

Щоб забезпечити адекватну роботу фільтра нижніх частот необхідно щоб значення коефіцієнта a знаходилось у межах $|a| < 1$.

**ПРИКЛАДИ ОФОРМЛЕННЯ БІБЛІОГРАФІЧНОГО ОПИСУ
РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ**

(згідно з ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання.
Загальні положення та правила складання» з урахуванням правок
(код УКНД 01.140.40))

Характеристика джерела	Приклад оформлення
Книги	
Один автор	1. Андріяш В. Державна етнополітика України в умовах глобалізації. Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2013. 328 с. 2. Краснова М. В. Договори в екологічному праві України : навч. посіб. / Київ. нац. ун-т ім. Тараса Шевченка. Київ : Алерта, 2012. 216 с. 3. Романюк А. Порівняльний аналіз політичних систем країн Західної Європи: інституційний вимір. Львів : Тріада плюс, 2004. 392 с. 4. Федорова Л. Д. З історії пам'яткоохоронної та музейної справи у Наддніпрянській Україні. 1870-ті-1910-ті рр. Київ, 2013. 373 с.
Два автори	1. Батракова Т. І., Калюжна Ю. В. Банківські операції : навч. посіб. Запоріжжя : ЗНУ, 2017. 130 с. 2. Богма О. С., Кисильова І. Ю. Фінанси : конспект лекцій. Запоріжжя : ЗНУ, 2016. 102 с. 3. Васильєв С. В., Ніколенко Л. М. Доказування та докази у господарському процесі України : монографія. Харків : Еспада, 2004. 192 с.
Три автори	1. Комаров В. В., Світлична Г. О., Удальцова І. В. Окреме провадження : монографія / за ред. В. В. Комарова. Харків : Право, 2011. 312 с. 2. Якобчук В. П., Богоявленська Ю. В., Тищенко С. В. Історія економіки та економічної думки : навч. посіб. Київ : ЦУЛ, 2015. 476 с. 3. Zhovinsky E.Ya., Kryuchenko N.O., Paparyha P.S. Geochemistry of Environmental Objects of the Carpathian Biosphere Reserve. Kyiv, 2013. 100 p.
Чотири і більше авторів	1. Прилипко С. М., Ярошенко О. М., Мороз С. В., Малиновська К. А. Укладення трудового договору: теоретико-прикладне дослідження : монографія. Харків : Юрайт, 2013. 288 с. 2. Основи охорони праці : підручник / О. І. Запорожець та ін. 2-ге вид. Київ : ЦУЛ, 2016. 264 с. 3. Клименко М. І., Панасенко Є. В., Стреляєв Ю. М., Ткаченко І. Г. Варіаційне числення та методи оптимізації : навч. посіб. Запоріжжя : ЗНУ, 2015. 84 с. 4. The mutual fund industry: Competition and investor welfare / R. G. Hubbard et. al. New York, NY : Columbia University Press, 2010. 256 p.
Автор(и) та редактор(и) /упорядники	1. Березенко В. В. PR як сфера наукового знання : монографія / за заг. наук. ред. В. М. Манакіна. Запоріжжя : ЗНУ, 2015. 362 с. 2. Грошевий Ю. М. Вибрані праці / упоряд.: О. В. Капліна, В. І. Маринів. Харків : Право, 2011. 656 с. 3. Експлуатація і технічне обслуговування газорозподільчих станцій магістральних газопроводів / заг. ред. А. А. Руднік. Київ, 2003. 370 с.

Без автора	<ol style="list-style-type: none"> 1. Антологія української літературно-критичної думки першої половини ХХ століття / упоряд. В. Агеєва. Київ : Смолоскип, 2016. 904 с. 2. Експлуатація і технічне обслуговування газорозподільчих станцій магістральних газопроводів / заг. ред. А. А. Руднік. Київ, 2003. 370 с. 3. Політологічний енциклопедичний словник / упоряд. В. П. Горбатенко. 2-ге вид., переробл. і допов. Київ : Генеза, 2004. 736 с. 4. Софія Київська: Візантія. Русь. Україна. Вип. II. Київ, 2012. 464 с. 5. Twenty-four hours a day. Miami, FL : BN Publishing, 2010. 400 p.
Багатотомні видання	<ol style="list-style-type: none"> 1. Енциклопедія історії України : у 10 т. / ред. рада: В. М. Литвин (голова) та ін. ; НАН України, Ін-т історії України. Київ : Наук. думка, 2005. Т. 9. 944 с. 2. Закалюк А. П. Курс сучасної української кримінології: теорія і практика : у 3 кн. Київ : Ін Юре, 2007. Кн. 1 : Теоретичні засади та історія української кримінології. 424 с. 3. Ушинський К. Д. Людина як предмет виховання. Спроба педагогічної антропології : вибр. твори. Київ : Рад. шк., 1983. Т. 1. 480 с.
Частина видання	
Книги	<ol style="list-style-type: none"> 1. Алексєєв В. М. Правовий статус людини та його реалізація у взаємовідносинах держави та суспільства в державному управлінні в Україні. <i>Теоретичні засади взаємовідносин держави та суспільства в управлінні</i> : монографія. Чернівці, 2012. С. 151–169. 2. Коломоєць Т. О. Адміністративна деліктологія та адміністративна деліктність. <i>Адміністративне право України</i> : підручник / за заг. ред. Т. О. Коломоєць. Київ, 2009. С. 195–197. 3. Наумов М. С. Напрями впливу інтелектуалізації економіки на розвиток ринкових відносин в Україні. <i>Трансформаційні процеси в суспільстві в умовах інформаційної економіки</i> : монографія / В. П. Решетило, М. С. Наумов, Ю. В. Федотова ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Харків, 2014. С. 213–241. 4. Саблук П. Т. Напрямки розвитку економіки в аграрній сфері виробництва. <i>Основи аграрного підприємництва</i> / за ред. М. Й. Маліка. Київ, 2000. С. 5–15.
Тези доповідей, матеріали конференцій	<ol style="list-style-type: none"> 1. Антонович М. Жертви геноцидів першої половини ХХ століття: порівняльно-правовий аналіз. <i>Голодомор 1932-1933 років: втрати української нації</i> : матеріали міжнар. наук.-практ. конф., м. Київ, 4 жовт. 2016 р. Київ, 2017. С. 133–136. 2. Анциперова І. І. Історико-правовий аспект акту про бюджет. <i>Дослідження проблем права в Україні очима молодих вчених</i> : тези доп. всеукр. наук.-практ. конф. (м. Запоріжжя, 24 квіт. 2014 р.). Запоріжжя, 2014. С. 134–137. 3. Зінчук Т. О. Економічні наслідки впливу продовольчих органічних відходів на природні ресурси світу. <i>Органічне виробництво і продовольча безпека</i> : зб. матеріалів доп. учасн. II Міжнар. наук.-практ. конф. Житомир : Полісся, 2014. С. 103–108. 4. Микитів Г. В., Кондратенко Ю. Позатекстові елементи як засіб формування медіакультури читачів науковопопулярних журналів. <i>Актуальні проблеми медіаосвіти в Україні та світі</i> : зб. тез доп. міжнар. наук.-практ. конф., м. Запоріжжя, 3-4 берез. 2016 р. Запоріжжя, 2016. С. 50–53.

<p>Статті із продовжуваних та періодичних видань</p>	<ol style="list-style-type: none"> 1. Кобильник В. Порівняльний метод як основа політологічного дослідження. <i>Збірник наукових праць «Політологічні студії»</i>. 2011. № 2. С. 54–65. 2. Коломоець Т. О. Оцінні поняття в адміністративному законодавстві України: реалії та перспективи формулювання їх застосування. <i>Вісник Запорізького національного університету. Юридичні науки</i>. Запоріжжя, 2017. № 1. С. 36–46. 3. Загірняк М., Костенко А. Про користування можливостями міжнародної бази даних Scopus. <i>Вища школа</i>. 2017. № 5–6. С. 48–55. 4. Кармазіна М., Могилевець О. Становлення і розвиток порівняльної методології в політичних дослідженнях. <i>Політичний менеджмент</i>. 2006. № 5. С. 3–17. 5. Коваль Л. Плюси і мінуси дистанційної роботи. <i>Урядовий кур'єр</i>. 2017. 1 листоп. (№ 205). С. 5. 6. Біленчук П., Обіход Т. Небезпеки ядерної злочинності: аналіз вітчизняного і міжнародного законодавства. <i>Юридичний вісник України</i>. 2017. 20-26 жовт. (№ 42). С. 14–15. 7. Bletskan D. I., Glukhov K. E., Frolova V. V. Electronic structure of 2H-SnSe₂: ab initio modeling and comparison with experiment. <i>Semiconductor Physics Quantum Electronics & Optoelectronics</i>. 2016. Vol. 19, No 1. P. 98–108. 8. Dalton R.J. Comparative politics of the industrial democracies: from the Golden Age to island hopping. <i>Political Science</i>. 1991. № 2. P. 15–43.
<p>Інші видання</p>	
<p>Законодавчі та нормативні документи</p>	<ol style="list-style-type: none"> 1. Конституція України : офіц. текст. Київ : КМ, 2013. 96 с. 2. Про освіту : Закон України від 05.09.2017 р. № 2145-VIII. <i>Голос України</i>. 2017. 27 верес. (№ 178-179). С. 10–22. 3. Деякі питання стипендіального забезпечення : Постанова Кабінету Міністрів України від 28.12.2016 р. № 1050. <i>Офіційний вісник України</i>. 2017. № 4. С. 530–543. 4. Про Концепцію вдосконалення інформування громадськості з питань євроатлантичної інтеграції України на 2017-2020 роки : Указ Президента України від 21.02.2017 р. № 43/2017. <i>Урядовий кур'єр</i>. 2017. 23 лют. (№ 35). С. 10. 5. Про затвердження Вимог до оформлення дисертації : наказ Міністерства освіти і науки від 12.01.2017 р. № 40. <i>Офіційний вісник України</i>. 2017. № 20. С. 136–141. 6. Інструкція щодо заповнення особової картки державного службовця : затв. наказом Нац. агентства України з питань Держ. служби від 05.08.2016 р. № 156. <i>Баланс-бюджет</i>. 2016. 19 верес. (№ 38). С. 15–16.
<p>Архівні документи</p>	<ol style="list-style-type: none"> 1. Лист Голови Співки «Чорнобиль» Г. Ф. Лепіна на ім'я Голови Ради Міністрів УРСР В. А. Масола щодо реєстрації Статуту Співки та сторінки Статуту. 14 грудня 1989 р. <i>ЦДАГО України</i> (Центр. держ. архів громад. об'єднань України). Ф. 1. Оп. 32. Спр. 2612. Арк. 63, 64 зв., 71. 2. Матеріали Ради Народних комісарів Української Народної Республіки. <i>ЦДАВО України</i> (Центр. держ. архів вищ. органів влади та упр. України). Ф. 1061. Оп. 1. Спр. 8–12. Копія; Ф. 1063. Оп. 3. Спр. 1–3. 3. Наукове товариство ім. Шевченка. <i>Львів. наук. б-ка ім. В. Стефаніка НАН України</i>. Ф. 1. Оп. 1. Спр. 78. Арк.1–7.

Патенти	<p>1. Спосіб лікування синдрому дефіциту уваги та гіперактивності у дітей: пат. 76509 Україна. № 2004042416; заявл. 01.04.2004; опубл. 01.08.2006, Бюл. № 8 (кн. 1). 120 с.</p> <p>2. Люмінісцентний матеріал: пат. 25742 Україна: МПК6 C09K11/00, G01T1/28, G21H3/00. № 200701472; заявл.12.02.07; опубл. 27.08.07, Бюл. № 13. 4 с.</p>
Препринти	<p>1. Панасюк М. І., Скорбун А. Д., Сплошной Б. М. Про точність визначення активності твердих радіоактивних відходів гамма-методами. Чорнобиль : Ін-т з проблем безпеки АЕС НАН України, 2006. 7, [1] с. (Препринт. НАН України, Ін-т проблем безпеки АЕС; 06-1).</p>
Стандарти	<p>1. ДСТУ 7152:2010. Видання. Оформлення публікацій у журналах і збірниках. [Чинний від 2010-02-18]. Вид. офіц. Київ, 2010. 16 с. (Інформація та документація).</p> <p>2. ДСТУ ISO 6107-1:2004. Якість води. Словник термінів. Частина 1 (ISO 6107-1:1996, IDT). [Чинний від 2005-04-01]. Вид. офіц. Київ : Держспоживстандарт України, 2006. 181 с.</p> <p>3. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила (ISO 4:1984, NEQ; ISO 832:1994, NEQ). [На заміну ДСТУ3582-97; чинний від 2013-08-22]. Вид. офіц. Київ : Мінекономрозвитку України, 2014. 15 с. (Інформація та документація).</p>
Каталоги	<p>1. Історико-правова спадщина України : кат. вист. / Харків. держ. наук. б-ка ім. В. Г. Короленка; уклад.: Л. І. Романова, О. В. Землянщина. Харків, 1996. 64 с.</p> <p>3. Пам'ятки історії та мистецтва Львівської області : кат.-довід. / авт.-упоряд.: М. Зобків та ін. ; Упр. Культури Львів. облдержадмін., Львів. іст. музей. Львів : Новий час, 2003. 160 с.</p>
Бібліографічні покажчики	<p>1. Боротьба з корупцією: нагальна проблема сучасності : бібліогр. покажч. Вип. 2 / уклад.: О. В. Левчук, відп. За вип. Н. М. Чала ; Запорізький національний університет. Запоріжжя : ЗНУ, 2017. 60 с.</p> <p>2. Микола Лукаш : біобібліогр. покажч. / уклад. В. Савчин. Львів : Вид. центр ЛНУ ім. І. Франка, 2003. 356 с. (Українська біобібліографія; ч. 10).</p> <p>3. Чернівецький національний університет імені Юрія Федьковича в незалежній Україні : бібліогр. покажч. / уклад.: Н. М. Загородна та ін.; наук. ред. Т. В. Марусик; відп. за вип. М. Б. Зушман. Чернівці : Чернівецький національний університет, 2015. 512 с. (До 140-річчя від дня заснування).</p>
Електронні ресурси	<p>1. Влада очима історії : фотовиставка. URL: http://www.kmu.gov.ua/control/uk/photogallery/gallery?galleryId=15725757& (дата звернення: 15.11.2017).</p> <p>2. Мар'їна О. Контент-стратегія бібліотек у цифровому середовищі <i>Бібліотечний вісник</i>. 2016. № 4. С. 8–12. URL: http://nbuv.gov.ua/UJRN/bv_2016_4_4 (дата звернення: 26.09.2017).</p> <p>3. Ганзенко О. О. Основні напрями подолання правового нігілізму в Україні. <i>Вісник Запорізького національного університету. Юридичні науки</i>. Запоріжжя, 2015. № 3. С. 20–27. URL: http://ebooks.znu.edu.ua/files/Fakhoviydannya/vznu/juridichni/VestUr2015v3/5.pdf. (дата звернення: 15.11.2017).</p>

	<p>4. Що таке органічні продукти і чим вони кращі за звичайні? <i>Екологія життя</i> : веб-сайт. URL: http://www.eco-live.com.ua/ (дата звернення: 12.10.2017).</p>
Автореферати дисертацій	<p>1. Гнатенко Н. Г. Групи інтересів у Верховній Раді України: сутність і роль у формуванні державної політики : автореф. дис. ... канд. політ. наук : 23.00.02. Київ, 2017. 20 с.</p> <p>2. Кравчук В. М. Припинення корпоративних правовідносин в господарських товариствах: автореф. дис. ... д-ра юрид. наук : 12.00.03 Харків, 2010. 36 с.</p> <p>3. Старовойт С. В. Видавнича діяльність Національної академії наук України у 1918–1933 рр. : автореф. дис. ... канд. іст. наук. Київ, 2003. 20 с.</p>
Дисертації	<p>1. Євдоченко О.О. Європейське бізнес-середовище в розвитку міжнародної економічної діяльності : дис... канд. екон. наук : 08.05.01 / Київський національний економічний ун-т. Київ, 2005. 235 с.</p> <p>2. Левчук С. А. Матриці Гріна рівнянь і систем еліптичного типу для дослідження статичного деформування складених тіл : дис. ... канд. фіз.-мат. наук : 01.02.04. Запоріжжя, 2002. 150 с.</p> <p>3. Вініченко О. М. Система динамічного контролю соціально-економічного розвитку промислового підприємства : дис. ... д-ра екон. наук : 08.00.04. Дніпро, 2017. 424 с.</p> <p>4. Bryant B. D. A sequentially articulated experiment to compare two instructional software input infrastructures: Doctoral dissertation / University at Albany. Albany, NY, 1998. 150 p.</p>