

Національний лісотехнічний університет України  
(повне найменування вишого навчального закладу)

Навчально-науковий інститут комп'ютерних наук  
та інформаційних технологій  
(повне найменування інституту, назва факультету(відділення))

Кафедра інформаційних систем та комп'ютерного моделювання  
(повна назва кафедри (предметної циклової комісії))

## **Пояснювальна записка**

до дипломної роботи  
перший (бакалаврський)  
(рівень вищої освіти)

на тему: «Розроблення системи автоматизованого проектування  
мультифункціональної шафи засобами AutoCAD»

Виконала студентка 4 курсу, групи ІСТ-41  
спеціальності:

126 „Інформаційні системи та технології”  
(шифр і назва напрямку підготовки спеціальності)

Гренюх Вікторія Ігорівна  
(прізвище, ім'я, по батькові)

Керівник: Сторожук О.Л.  
(прізвище, ініціали)

Рецензент: Яцишин С.І.  
(прізвище, ініціали)

Львів-2025

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій

Кафедра інформаційних систем та комп'ютерного моделювання

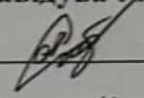
Рівень вищої освіти перший (бакалаврський)

Спеціальність 126 «Інформаційні системи та технології»

(шифр і назва)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри ІСКМ**

 Сторожук О.Л.  
« 15 » 11 2024 року

**ЗАВДАННЯ**

**НА ДИПЛОМНУ РОБОТУ СТУДЕНТЦІ**

Гренюх Вікторії Ігорівні

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи автоматизованого проєктування мультифункціональної шафи засобами AutoCAD.

Керівник роботи Сторожук Олександр Леонідович, кандидат технічних наук, доцент.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від «15» листопада 2024 року № С-884

2. Термін подання студентом роботи 12 червня 2025 року

3. Вихідні дані до Розробити систему автоматизованого проєктування мультифункціональної шафи в середовищі AutoCAD з використанням мови програмування AutoLISP та технологій DCL/VisualLISP.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) 1) Стан проблемної області; 2) Інформаційне та математичне забезпечення; 3) Програмне та технічне забезпечення;

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

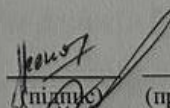
Презентація до диплому

6. Дата видачі завдання 18 листопада 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних даних та інших джерел згідно досліджуваної теми	19.11.2024р. - 03.12.2024р.	Виконано
2	Аналіз досліджуваної теми та вибір відповідних варіантів її розробки	04.12.2024р. - 18.12.2024р.	Виконано
3	Постановка задачі та її формалізація	19.12.2024р. - 09.01.2025р.	Виконано
4	Вибір та обґрунтування методів і засобів проведення дослідження	10.01.2025р. - 31.01.2025р.	Виконано
5	Розроблення концептуальної схеми реалізації завдання	01.02.2025р. - 21.02.2025р.	Виконано
6	Програмна реалізація завдання	22.02.2025р. - 31.03.2025р.	Виконано
7	Тестування програмного продукту та отриманих результатів	01.04.2025р. - 20.04.2025р.	Виконано
8	Розробка пояснювальної записки бакалаврської роботи	21.04.2025р. - 15.05.2025р.	Виконано
9	Коригування пояснювальної записки згідно вимог, розроблення презентації	16.05.2025р. - 20.05.2025р.	Виконано

Студентка

  
(підпис)

Гренюх В.І.

(прізвище та ініціали)

Керівник роботи

  
(підпис)

Сторожук О.Л.

(прізвище та ініціали)

## **АНОТАЦІЯ**

Дипломна робота містить 50 сторінок пояснювальної записки, 13 рисунків, 7 формул, 1 додатку, 15 джерел.

У межах дипломної роботи створено програмне забезпечення для автоматизованого проєктування мультифункціональної шафи в середовищі AutoCAD. Система дозволяє вводити геометричні параметри конструкції, перевіряти їх на коректність і будувати тривимірну модель виробу. Програму реалізовано засобами VisualLISP та DCL, з інтеграцією у вигляді VLX-додатку для AutoCAD 2025. Особливу увагу приділено перевірці вхідних даних, що дозволяє уникати помилок при побудові моделі та забезпечує точність розрахунків і проєктування.

Ключові слова: AutoCAD, AutoLISP, DCL, проєктування меблів, 3D-моделювання, САПР.

## **ABSTRACT**

The thesis contains 50 pages of explanatory notes, 13 figures, 7 formulas, 1 appendix, and 15 sources.

As part of the thesis, software was created for the automated design of a multifunctional cabinet in the AutoCAD environment. The system allows you to enter the geometric parameters of the structure, check them for correctness, and build a three-dimensional model of the product. The programme is implemented using VisualLISP and DCL, with integration in the form of a VLX add-on for AutoCAD 2025. Particular attention is paid to checking the input data, which allows avoiding errors when building the model and ensures the accuracy of calculations and design.

Keywords: AutoCAD, AutoLISP, DCL, furniture design, 3D modelling, CAD.

## ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити програмне забезпечення автоматизованого проектування мультифункціональної шафи. Програмне забезпечення розробити у вигляді додатка для AutoCAD 2025, з наступними функціональними можливостями:

- ввід геометричних розмірів шафи;
- валідація типу введених даних (перевірка, що введене значення є числовим);
- виведення результатів геометричних розрахунків у файл для подальшого створення необхідної документації.

Програмне забезпечення реалізувати за допомогою мов програмування VisualLisp та DCL в середовищі VisualLISP for AutoCAD, з наглядним та зрозумілим графічним інтерфейсом.

Розроблені LSP та DCL – файли об'єднати у VLX-додаток.

## ЗМІСТ

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ .....	10
1.1 Загальний огляд меблевого виробництва .....	10
1.2 Аналіз особливостей проектування корпусних меблів .....	11
1.3 Огляд існуючих САПР для проектування меблів.....	12
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....	16
2.1 Виділення об'єктів дослідження .....	16
2.2 Побудова дерева проблем та дерева цілей .....	17
2.3 Специфікація вимог до програмного забезпечення.....	19
2.3.1 Характеристики розроблюваного продукту .....	20
2.3.2 Середовище функціонування .....	21
2.3.3 Характеристика функціональних компонентів системи.....	22
2.4 Структурна схема системи із урахуванням інформаційних потоків .....	23
2.5 Математичне та алгоритмічне забезпечення .....	25
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	31
3.1 Основні функціональні можливості AutoCAD .....	31
3.2 Обґрунтування вибору програмного забезпечення та інструментів.....	33
3.3 Реалізація програмного забезпечення.....	34
3.4 Інтерфейс користувача та взаємодія .....	41
3.5 Формування результатів.....	44
3.6 Особливості реалізації обробки кромки у деталях шафи та тумбочки .....	46
ВИСНОВКИ .....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ .....	53
ДОДАТОК А. КОД ПРОГРАМИ .....	53

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

AutoCAD – система автоматизованого проєктування для створення 2D і 3D моделей, з можливістю програмної автоматизації через AutoLISP.

САПР – система автоматизованого проєктування;

AutoLISP – спеціалізований діалект мови LISP, розширений наборами функцій, що дозволяють працювати з об'єктами, таблицями та словниками середовища AutoCAD.

ОС – операційна система;

ПЗ – програмне забезпечення;

ЧПК – числове програмне керування;

DCL (Dialog Control Language) – мова для створення діалогових вікон в AutoCAD, що забезпечує зручну взаємодію користувача з AutoLISP-програмами.

## ВСТУП

У сучасному світі, де ефективність і персоналізація стали ключовими факторами в дизайні інтер'єрів, меблеве виробництво дедалі більше інтегрує цифрові технології в усі етапи свого циклу - від ідеї до готового виробу. Особливої актуальності набуває впровадження САПР, які дають змогу скоротити час створення моделей, мінімізувати похибки та адаптувати конструкції під індивідуальні вимоги замовника.

Однією з найпотужніших і найгнучкіших платформ для реалізації таких рішень є AutoCAD – це програмне середовище, яке поєднує інструменти 2D- та 3D-моделювання з можливістю розширення функціоналу за допомогою мов програмування AutoLISP, VisualLISP та DCL. Це відкриває широкі можливості для розробки інтелектуальних рішень, орієнтованих на конкретні виробничі задачі.

У рамках даного дипломного проєкту створено систему автоматизованого проєктування мультифункціональної шафи - універсального меблевого виробу, який поєднує кілька функціональних блоків: основний корпус, тумбу та декоративну арку. Система дозволяє гнучко задавати параметри виробу, генерувати тривимірну модель у середовищі AutoCAD, а також отримувати розрахункові розміри деталей для подальшого виготовлення. Таким чином, запропоноване рішення спрямоване на автоматизацію та оптимізацію процесу створення меблів на індивідуальне замовлення.

**Актуальність теми** зумовлена необхідністю прискорення процесу проєктування нестандартних меблів у невеликих виробництвах та студіях індивідуального дизайну, де економія часу і точність відіграють критичну роль.

**Об'єкт дослідження** – процес автоматизованого проєктування меблів у середовищі AutoCAD.

**Предмет дослідження** – реалізація параметричної побудови корпусних меблів із використанням AutoLISP, VisualLISP та DCL для створення користувацького інтерфейсу введення параметрів.

**Мета роботи** – створити інтерактивну систему для автоматизованого проєктування багатофункціональної шафи у середовищі AutoCAD із можливістю змінювати конфігурацію виробу відповідно до заданих розмірів.

**Наукова новизна** полягає у розробці алгоритмів динамічного формування моделі меблів за допомогою мови програмування AutoLISP, що дозволяє інтегрувати перевірку параметрів і логіку побудови безпосередньо у робоче середовище AutoCAD.

**Практичне значення** розробленого додатку полягає в можливості значного прискорення процесу проєктування, отримання точних креслень і специфікацій, а також зменшення кількості помилок на етапі виробництва.

## РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1 Загальний огляд меблевого виробництва

Меблева промисловість є однією з важливих галузей виробництва, що охоплює різні сегменти продукції для житлових і громадських інтер'єрів. Серед них особливо виділяються корпусні меблі – шафи, комоди, стелажі, кухонні гарнітури та інші виробі з жорстким каркасом. За даними статистики, виготовлення корпусних меблів є одним з найбільш затребуваних напрямів меблевого бізнесу – близько 25% усього меблевого виробництва припадає саме на цей сегмент. Попит на корпусні меблі залишається стабільним, особливо у зв'язку з облаштуванням нових осель, де потрібні функціональні системи зберігання одягу, книг, посуду тощо. Сучасний споживач добре розуміє призначення корпусних меблів і висуває високі вимоги до їх якості, функціональності та довговічності. Важливими чинниками конкурентоспроможності корпусних меблевих виробів є ергономічність конструкції, використання якісних матеріалів та фурнітури, а також привабливий дизайн.

Поряд із корпусними меблями набули широкого поширення модульні меблі – набори окремих модулів, виконані в єдиному стилі, які можна комбінувати та розміщувати на власний розсуд. Іншими словами, модульні системи являють собою конструктор для створення індивідуального інтер'єру: споживач може вибрати необхідні елементи (шафи, полиці, тумби тощо) і скомпонувати їх оптимальним чином для свого приміщення. Перевагою модульних меблів є гнучкість та адаптивність до простору – їх можна легко переналаштувати чи доповнити новими елементами при зміні потреб або переїзді. За рахунок взаємозамінності секцій забезпечується масштабованість системи, що важливо для росту потреб людей чи бізнесу. Згідно з фаховими оглядами, модульні та трансформовані меблі сьогодні розглядаються як тренд, оскільки дозволяють раціонально використовувати простір, не жертвуючи комфортом і стилем. Це особливо актуально для малогабаритних квартир, де мультифункціональні рішення (наприклад, шафа-купе з вбудованим

робочим місцем або відкидним спальним місцем) допомагають поєднати кілька функцій в одному предметі меблів.

Таким чином, сучасне меблеве виробництво зорієнтоване на виготовлення корпусних та модульних меблів, які відповідають вимогам ергономіки та практичності. Підприємства галузі впроваджують нові матеріали (ламіновані ДСП, МДФ, пластики, скло тощо) та фурнітуру (напрявні, завіси з доводчиками, системи трансформації) задля підвищення якості і функціональності виробів. Одночасно зростає роль інформаційних технологій у проектуванні та виробництві меблів – ПЗ CAD/CAM, системи автоматизованого проектування і підготовки виробництва сприяють скороченню циклу розробки нових моделей і підвищенню точності виготовлення деталей. Особливо актуальним є використання спеціалізованих САПР для корпусних меблів, що спрощують процес дизайну, розкладки деталей на матеріалі та генерації конструкторської документації [1].

## **1.2 Аналіз особливостей проектування корпусних меблів**

Проектування корпусних меблів, зокрема мультифункціональних шаф, потребує врахування комплексу чинників, які визначають ергономічність, міцність, функціональність і відповідність нормативам. Ергономіка є ключовим елементом, що забезпечує зручне та безпечне використання виробу. При формуванні внутрішньої структури шафи необхідно враховувати антропометричні дані користувачів: висоту розміщення полиць, глибину для зберігання одягу, радіуси досяжності рук, кут відкривання дверей тощо. Шафа повинна бути зручною як для дітей, так і для дорослих користувачів різного зросту й фізичних можливостей.

Окрім ергономіки, дотримання державних стандартів (ДСТУ, ГОСТ) є обов'язковим при проектуванні. Наприклад, згідно ГОСТ 19882-91 і ГОСТ 16371-2014, меблі мають витримувати задані навантаження, бути безпечними в експлуатації, а матеріали – відповідати вимогам нетоксичності, вологостійкості та зносостійкості. Особливу увагу слід приділити покриттю кромки (особливо для ДСП) задля зменшення виділення формальдегідів [2].

Що стосується конструкції, вона має бути максимально простою, міцною та технологічною. Це означає, що товщина панелей вибирається відповідно до очікуваного навантаження, вузли з'єднань повинні забезпечувати жорсткість і тривалу експлуатацію, а використання типових деталей і отворів значно спрощує виробництво й ремонт. Уніфіковані елементи дозволяють прискорити збирання і знизити витрати на виготовлення. Важливо, щоб шафа могла бути легко розібрана і зібрана повторно без втрати якості.

Надійність – це ще один критичний аспект. Конструкція повинна зберігати форму і функціональність протягом тривалого часу, не допускаючи провисання полиць, перекосу фасадів, розхитування кріплень. Для цього передбачають запас міцності та використовують фурнітуру високої якості. Матеріали мають бути не лише міцними, а й безпечними для здоров'я, легко очищуваними й довговічними [3].

Таким чином, ефективне проєктування корпусних меблів передбачає узгодженість між вимогами нормативів, ергономіки, технічної реалізації та експлуатаційної зручності. Це формує основу для створення автоматизованої САД-системи, яка мінімізує типові помилки ручного проєктування та підвищує точність кінцевого виробу.

### **1.3 Огляд існуючих САПР для проєктування меблів**

Для підвищення ефективності розробки меблів сьогодні використовуються різноманітні САПР, спеціалізовані саме на меблевих виробках. Вони дозволяють конструктору швидко створити тривимірну модель меблів, автоматично отримати необхідні креслення, специфікації деталей, карти розкрою та іншу документацію. Нижче розглянуто основні такі програмні продукти, їх особливості та можливості.

**AutoCAD** – одна з найпотужніших і найпоширеніших систем САПР загального призначення, що широко використовується і в меблевому проєктуванні. AutoCAD дозволяє створювати точні 2D-креслення й детальні 3D-моделі з високим рівнем точності. Однією з ключових переваг є підтримка мови AutoLISP, яка дозволяє автоматизувати рутинні дії, створювати користувацькі команди й налаштовувати

індивідуальні функції, що особливо корисно при серійному або параметричному моделюванні меблів. Завдяки широким можливостям для створення блоків, шарів, динамічних елементів, а також вбудованим засобам візуалізації й стилізації креслень, AutoCAD надає потужні інструменти не лише для інженерного проєктування, а й для презентації концептів. Програма підтримує інтеграцію з іншими CAD/CAM-платформами, імпорт/експорт файлів, роботу з шаблонами та базами даних, що робить її універсальним інструментом для професійного середовища.

**SolidWorks** – професійне інженерне САПР для 3D-моделювання, що часто використовується у складному меблевому проєктуванні. Завдяки можливостям твердотільного моделювання, аналізу складання, різноманітним дослідженням та автоматичного створення креслень, SolidWorks забезпечує високий рівень точності й контроль за конструктивними елементами. Також підтримує параметричне проєктування і розрахунок міцності, що важливо при створенні багатфункціональних меблевих систем. Однією з ключових переваг є зручність керування змінами: будь-які модифікації в одній частині моделі автоматично враховуються в усіх пов'язаних елементах і кресленнях. SolidWorks підтримує створення бібліотек стандартних компонентів, інтеграцію з CAM/CAE-системами, а також візуалізацію моделей для презентацій або клієнтських демонстрацій. Завдяки цьому, програма активно використовується як інженерами-конструкторами, так і виробниками меблів для контролю якості, узгодження виробничих процесів та інтеграції з ЧПК-обладнанням.

**3ds Max** – потужне ПЗ для 3D-візуалізації та анімації, яке активно використовується у меблевому та інтер'єрному проєктуванні. Основною перевагою є високий рівень фотореалістичної візуалізації, що дозволяє точно відтворити вигляд меблів, текстури матеріалів, освітлення та тіні. 3ds Max підтримує імпорт моделей з інших CAD-систем, а також має можливість інтеграції з бібліотеками готових меблевих елементів, що значно скорочує час на створення інтер'єру. Завдяки вбудованим механізмам анімації, камера-об'єктам та візуальним ефектам, 3ds Max є незамінним інструментом для створення презентаційних матеріалів, каталогів або демонстрацій для замовників. Хоча програма не призначена для створення робочих

креслень чи технічної документації, вона ідеально доповнює інженерні САПР, дозволяючи розширити візуальний аспект проектування.

**SketchUp** – інтуїтивно зрозумілий і гнучкий інструмент для 3D-моделювання, який часто використовується для створення меблів та інтер'єрів. SketchUp дозволяє швидко створювати візуальні концепції меблевих конструкцій, використовуючи як базові геометричні примітиви, так і готові моделі з бібліотеки 3D Warehouse. Хоча програма не призначена для глибокого інженерного моделювання або створення технічної документації, вона підтримує модулі для створення креслень, рендерінгу, а також розрахунку вартості проєктів. SketchUp також має гнучку систему плагінів, які розширюють її можливості: наприклад, розрахунок розкладки деталей, інтеграція з верстатами ЧПК або автоматичне формування специфікацій. Середовище моделювання підтримує масштабування, точні розміри та редагування у реальному часі. SketchUp особливо корисний для дизайнерів, які працюють над індивідуальними або клієнтськими замовленнями, де важлива швидка візуалізація і можливість гнучкого редагування конструкції.

**Астра Конструктор Меблів** – програма українського розробника для проектування корпусних меблів та дизайну інтер'єру. Вона надає зрозумілі та зручні інструменти для швидкого створення виробу, розміщення його в приміщенні і перегляду результату в 3D. На основі побудованої моделі система готує комплект креслень та звітів по проєкту і дозволяє експортувати дані деталей в замовлення для програми розкрою або безпосередньо на верстаті ЧПК. Фактично комплекс програм “Астра” складається з двох основних частин: Астра Конструктор для дизайн-проектування меблів та Астра Розкрій для оптимізації розкрою листових матеріалів. Взаємодія між ними забезпечує повний цикл: спроектований виріб з “Астра Конструктор” передається в “Астра Розкрій”, де автоматично формується оптимальна карта розкрою плит з мінімальними відходами, генеруються етикетки деталей і файли для верстатів з ЧПК. ПЗ “Астра” інтегрується з обліковими системами для управління виробництвом, що важливо для середніх та великих меблевих підприємств.

У першому розділі було проаналізовано особливості сучасного меблевого проектування, з акцентом на вимоги до ергономіки, конструкції, нормативних

стандартів і технологічних особливостей виробництва. Встановлено, що для досягнення високої якості та ефективності виготовлення корпусних меблів необхідне комплексне застосування цифрових інструментів.

Окрему увагу приділено огляду програмних САПР, які відіграють ключову роль у проєктуванні меблів. Проаналізовано функціональні можливості AutoCAD, SolidWorks, 3ds Max, SketchUp, Астра Конструктор Меблів. AutoCAD вирізняється своєю універсальністю, підтримкою мови AutoLISP для автоматизації процесів, потужними інструментами креслення та візуалізації. SolidWorks забезпечує високоточне параметричне моделювання, симуляцію збірок, розрахунки навантажень та повну інтеграцію з виробничими системами. SketchUp, у свою чергу, забезпечує гнучке 3D-моделювання і швидку візуалізацію концепцій, з можливістю розширення функцій через плагіни. 3ds Max надає інструменти фотореалістичної візуалізації, які доповнюють технічну частину проєкту презентаційною складовою. Астра Конструктор меблів включає в себе параметричну побудову меблів, функцію розкладки деталей на листах для розкрою, простий інтерфейс та пряма інтеграція з виробничим обладнанням є ефективним інструментом для підприємств.

Таким чином, результати розділу підтверджують доцільність створення спеціалізованої автоматизованої системи проєктування корпусних меблів на базі платформи AutoCAD з використанням мови програмування AutoLISP. Це дозволить підвищити точність технічної документації, спростити адаптацію під виробничі процеси та забезпечити гнучкість і масштабованість для індивідуальних і серійних замовлень.

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Виділення об'єктів дослідження

Розробка системи автоматизованого проєктування передбачає чітке визначення меж дослідження – тобто конкретних об'єктів, що підлягають моделюванню, аналізу та побудові в середовищі AutoCAD. Це дозволяє сфокусувати реалізацію лише на тих елементах, які входять до складу проєктованого виробу, забезпечити точність геометричних розрахунків і побудову валідної тривимірної моделі без надлишкової деталізації або узагальнень. Об'єктом проєктування є мультифункціональна шафа з додатковою тумбочкою, що складається з таких елементів:

- корпусу шафи, який формується з основних панелей: лівої, правої, задньої, верхньої та нижньої;
- горизонтальних полиць (верхня, середня, нижня), які поділяють внутрішній об'єм шафи, а також вертикальна перегородка для нижньої полиці;
- дверцята, які закривають шафу;
- арка, розташована у верхній частині шафи як декоративний елемент;
- тумбочка, що проєктується окремо і містить свій корпус, дверцята та полицю;
- кромкові обробки та зони обрізки, які формуються відповідно до параметрів, заданих користувачем.

Кожен з елементів має чітко визначені геометричні параметри, такі як ширина, висота, глибина, товщина матеріалу, відступи, координати вставки. Також враховуються взаємозв'язки між елементами – наприклад, полиці будуються з урахуванням положення попередніх, дверцята розраховуються відповідно до висоти отвору, а арка точно вирівнюється по центру верхнього фронтального краю.

Особливість автоматизованого проєктування полягає в тому, що всі ці елементи не просто змальовуються вручну, а генеруються програмно на основі введених числових значень. Відповідно, об'єкти дослідження є параметризованими конструктивними одиницями, які формуються згідно з алгоритмом і перевіряються на коректність введення даних.

Також для кожного об'єкта проєктування були створені окремі функції в AutoLISP, які враховують правила побудови 3D-паралелепіпеда з певними розмірами, внутрішні зміщення, які враховують товщину матеріалу, наявність або відсутність кромки на певних гранях, особливості вирівнювання, орієнтації в просторі та сполучення з іншими частинами виробу.

Таке детальне виділення об'єктів дослідження дозволяє систематизувати логіку побудови, реалізувати алгоритми перевірки і побудови без конфліктів та помилок, легко масштабувати систему для інших меблевих виробів.

Таким чином, виділення об'єктів дослідження є критично важливим кроком, що дозволяє узгодити інформаційну, алгоритмічну і візуальну моделі проєкту, а також забезпечити логічну цілісність на всіх етапах проєктування - від формування вхідних даних до побудови готової 3D-моделі.

## **2.2 Побудова дерева проблем та дерева цілей**

У рамках розробки інформаційно-алгоритмічного забезпечення було виконано структурований аналіз основних труднощів, які виникають у процесі ручного проєктування корпусних меблів. З цією метою побудовано дерево проблем (рис. 2.1), що дозволяє наочно відобразити причинно-наслідкові зв'язки між ключовими обмеженнями існуючого підходу та похідними від них помилками. Цей інструмент є важливою складовою концептуального проєктування, оскільки допомагає виявити вузлові проблеми, на які має бути спрямована автоматизація процесів. Центральна проблема полягає у високій складності та ймовірності помилок, притаманних ручному проєктуванню. Вона розгалужується на три основні проблеми:

**Головна проблема:** Високий рівень трудомісткості та можливість помилок при ручному проєктуванні корпусних меблів.

**Проблема 1.** Неточність геометричних розрахунків. Основними причинами цього є відсутність автоматизованого розрахунку габаритів деталей на основі вхідних даних та помилки при ручному введенні числових значень параметрів.

**Проблема 2.** Складність перевірки допустимості параметрів, що зумовлена ризиком введення некоректних даних у DCL-форму (наприклад, літер або символів замість чисел або чисел, що виходять за межі допустимого).

**Проблема 3.** Технічна обмеженість засобів проектування, до якої належать відсутність автоматизованого формування документації, неможливість інтеграції користувацьких розрахункових модулів без програмування, порушення сумісності параметрів між окремими модулями (наприклад, шафою й тумбочкою).

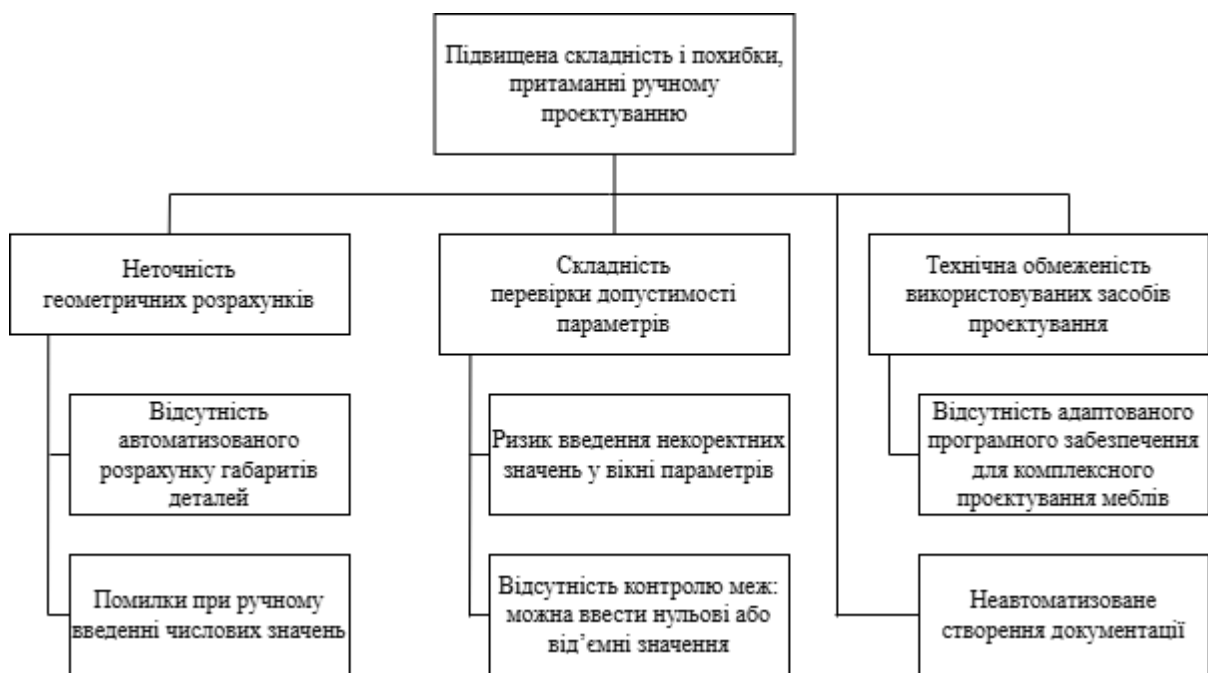


Рисунок 2.1 - Дерево проблем

На основі аналізу проблем була сформована система цілей, що логічно трансформує виявлені бар'єри в стратегічні завдання розробки. Побудоване дерево рішень (рис. 2.2) репрезентує ці цілі у вигляді тривірневої ієрархії, де центральна мета деталізується до конкретних реалізаційних кроків.

**Головна ціль:** створити автоматизовану систему проектування мультифункціональної шафи в середовищі AutoCAD, яка дозволить підвищити точність, пришвидшити розрахунки та мінімізувати кількість помилок.

Вона досягається через реалізацію таких напрямків:

**Напрямок 1.** Автоматизація геометричних розрахунків, що полягає у створенні алгоритмів розрахунку габаритів деталей відповідно до введених параметрів, а також застосуванні VisualLISP для побудови 3D-об'єктів на основі точних координат і розмірів.

**Напрямок 2.** Валідація параметрів, що відбувається за допомогою впровадження перевірки типу введених даних (обмеження введення лише чисел, які входять в межі дозволеного).

**Напрямок 3.** Розширення функціональності AutoCAD, що реалізується за допомогою створення інтерфейсу взаємодії користувача через DCL-форму з підстановкою заданих початкових значень, реалізацією автоматичного експорту розрахункових результатів у текстовий файл.

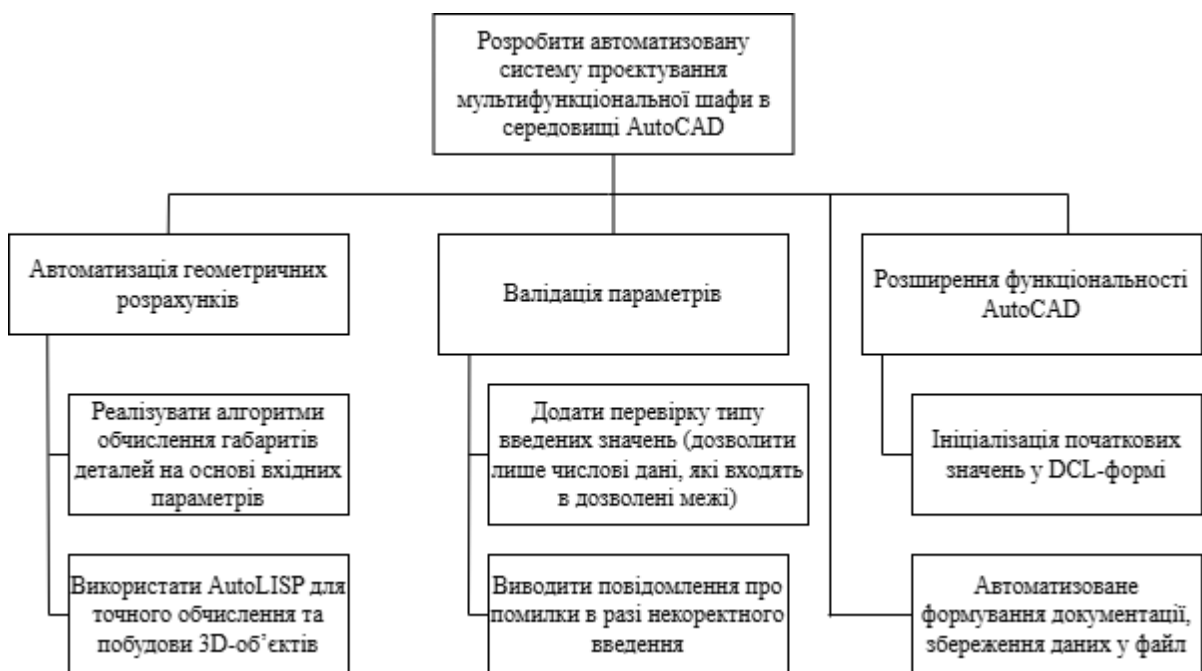


Рисунок 2.2 – Дерево рішень

## 2.3 Специфікація вимог до програмного забезпечення

Програмний модуль, що розробляється, призначений для автоматизованого проектування мультифункціональної шафи в середовищі AutoCAD. Основне призначення – зменшити обсяг рутинних операцій, підвищити точність

конструктивних рішень та забезпечити швидке отримання документації. Рішення реалізується у вигляді надбудови, яка інтегрується у AutoCAD і дозволяє користувачеві вводити параметри, перевіряти їх правильність, запускати програму, обчислити дані та зберігати результати у зручному форматі.

### **2.3.1 Характеристики розроблюваного продукту**

#### ***Функціональні можливості:***

1. Задання геометричних параметрів шафи та тумбочки (ширина, висота, глибина, висоти полиць тощо).
2. Автоматична валідація введених даних, включаючи розпізнавання типу (допускаються лише числові значення а також значення з обмеженими розмірами).
3. Обчислення розмірів усіх елементів (панелі, полиці, двері, арка тощо) відповідно до введених параметрів.
4. Генерація тривимірної моделі шафи і тумбочки в просторі AutoCAD з використанням команд побудови 3D-об'єктів.
5. Застосування графічного інтерфейсу користувача на основі DCL-форм для зручного введення та редагування даних.
6. Формування вихідних даних у вигляді текстового файлу з розрахунками, у двох варіантах - з урахуванням кромки та без.
7. Пакетне об'єднання компонентів (AutoLISP, VisualLISP, DCL) у єдиний VLX-додаток для AutoCAD.

#### ***Нефункціональні вимоги:***

1. Інтерфейс має бути логічно зрозумілим, з мінімальним порогом входження для користувача.
2. Система повинна забезпечувати швидку обробку та побудову моделі без затримок.

3. Підтримка AutoCAD 2025 та сумісність із наступними версіями (в межах backward compatibility).
4. Надійність: програма має обробляти некоректні дані без критичних помилок і аварійного завершення роботи.

### **2.3.2 Середовище функціонування**

#### ***ПЗ:***

1. AutoCAD 2025 (або новіші версії).
2. Середовище VisualLISP для реалізації та тестування AutoLISP-коду й форм DCL.

#### ***ОС:***

1. Microsoft Windows 10 або новіші версії.

#### ***Обладнання:***

1. Процесор: не нижче Intel Core i5.
2. Оперативна пам'ять: мінімум 8 ГБ.
3. Дисковий простір: не менше 10 ГБ для інсталяції AutoCAD і збереження проєктних файлів.
4. Відеокарта: 4 GB GPU, необхідна для візуалізації 3D-об'єктів.

#### ***Вимоги до користувача:***

1. Базове володіння AutoCAD.
2. Розуміння основ проєктування корпусних меблів.
3. Навички роботи з геометричними параметрами (розміри, пропорції, допустимі межі).

Цей опис визначає основні характеристики програмного забезпечення, що гарантують його відповідність очікуванням користувачів та умовам його застосування в реальному середовищі.

### **2.3.3 Характеристика функціональних компонентів системи**

#### ***Введення параметрів геометричних розмірів шафи та тумбочки***

##### *Опис і пріоритет*

Ключовий етап взаємодії користувача з системою, що передбачає введення основних габаритів корпусних елементів (висота, ширина, глибина тощо) через інтерактивну DCL-форму. Пріоритет цієї функції є високим, оскільки вона визначає подальший логічний ланцюг проектування.

##### *Послідовність дій*

1. Користувач запускає програму та відкриває форму введення параметрів.
2. У відповідні поля вводяться значення геометричних розмірів:
  - висота, ширина, глибина шафи;
  - аналогічні параметри для тумбочки.
3. Система виконує базову перевірку відповідності даних типу (тільки числа, які входять в межі дозволеного діапазону).
4. У разі помилки відображається повідомлення з підказкою для виправлення.

##### *Формування документації та збереження результатів*

Система генерує структурований текстовий файл із результатами проектування (розміри, габарити, назви деталей). Пріоритет середній - важливо для підготовки до виробництва.

#### ***Побудова 3D-моделі шафи та тумбочки***

##### *Опис та пріоритет*

Основна функція системи – генерація 3D-моделі у середовищі AutoCAD на основі параметрів, заданих користувачем. Пріоритет - критично високий.

##### *Послідовність дій*

1. Після введення параметрів система викликає відповідні функції AutoLISP.
2. Будується модель, що включає:
  - корпус;
  - внутрішні полиці, арку, перегородки;

- дверцята та тумбочку.
3. Модель відображається в робочому просторі AutoCAD.
  4. Користувач має змогу переглянути, обертати, масштабувати модель.

Ці функціональні характеристики забезпечують основний цикл роботи системи автоматизованого проєктування корпусних меблів та визначають логіку її реалізації у середовищі AutoCAD.

## **2.4 Структурна схема системи із урахуванням інформаційних потоків**

Структурна схема програмного забезпечення для автоматизованого проєктування мультифункціональної шафи та тумбочки в AutoCAD (рис. 2.3) відображає логічну організацію компонентів системи, характер їх взаємодії та потоки інформації між окремими модулями. Ця схема є концептуальним уявленням роботи системи, яка дозволяє краще зрозуміти внутрішню архітектуру рішення, послідовність обробки вхідних даних, механізми побудови 3D-моделі та формування технічної документації.

Процес проєктування розпочинається з роботи користувача у графічному інтерфейсі, реалізованому мовою DCL. Через цей інтерфейс користувач вводить геометричні параметри майбутньої конструкції - такі як висота, ширина, глибина шафи, параметри арки, кількість полиць, ті ж дані для тумбочки тощо. Введені дані автоматично передаються до модуля перевірки коректності, де здійснюється валідація – перевірка типів даних (чи є значення числовими і чи входить число в межі дозволеного).

У випадку виявлення помилок система повертає зворотний зв'язок до інтерфейсу – із повідомленням про помилку та рекомендаціями щодо виправлення. Якщо всі параметри пройшли перевірку, дані надходять до модуля побудови тривимірної моделі. Саме цей модуль, реалізований за допомогою AutoLISP та ActiveX, відповідає за автоматичне створення 3D-об'єкта у просторі моделі AutoCAD. На основі перевірених даних формуються корпус шафи, полиць, дверцята, арочні

елементи, а також компоненти тумбочки. Створена модель доступна для перегляду в реальному часі та подальших змін.

Після побудови моделі автоматично активується модуль генерації розрахункових даних, який здійснює обчислення габаритів усіх конструктивних деталей з урахуванням можливих кромок або без них. Отримані розміри та специфікації записуються у структурований текстовий файл формату .txt, придатний для технічної документації та передачі у виробництво.

Таким чином, система реалізує чітко структуровану послідовність інформаційних потоків. Для більшого розуміння було створено структурну схему взаємодій (рис 2.3). "GUI користувача" охоплює введення, перевірку та передачу параметрів на побудову. "Механізм перевірки введених значень" реалізується при виявленні помилок у введених значеннях. "Система формування 3D-моделі" передбачає побудову та вивід тривимірної конструкції в AutoCAD. "Механізм виводу параметрів конструкції" включає збереження обчислених даних у вигляді файлів для подальшого використання.

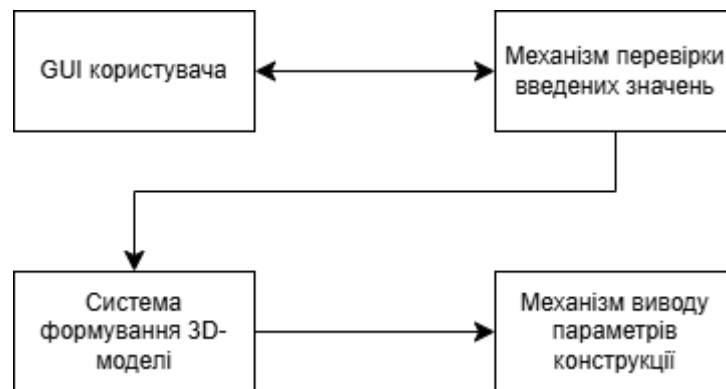


Рисунок 2.3 – Структурна схема системи із урахуванням інформаційних потоків

Діаграма прецедентів (рис. 2.4) демонструє функціональні сценарії взаємодії системи з основними акторами – технологом, дизайнером і користувачем. Ця діаграма показує ролі користувачів та межі системи, надаючи цілісне уявлення про її функціональне наповнення.

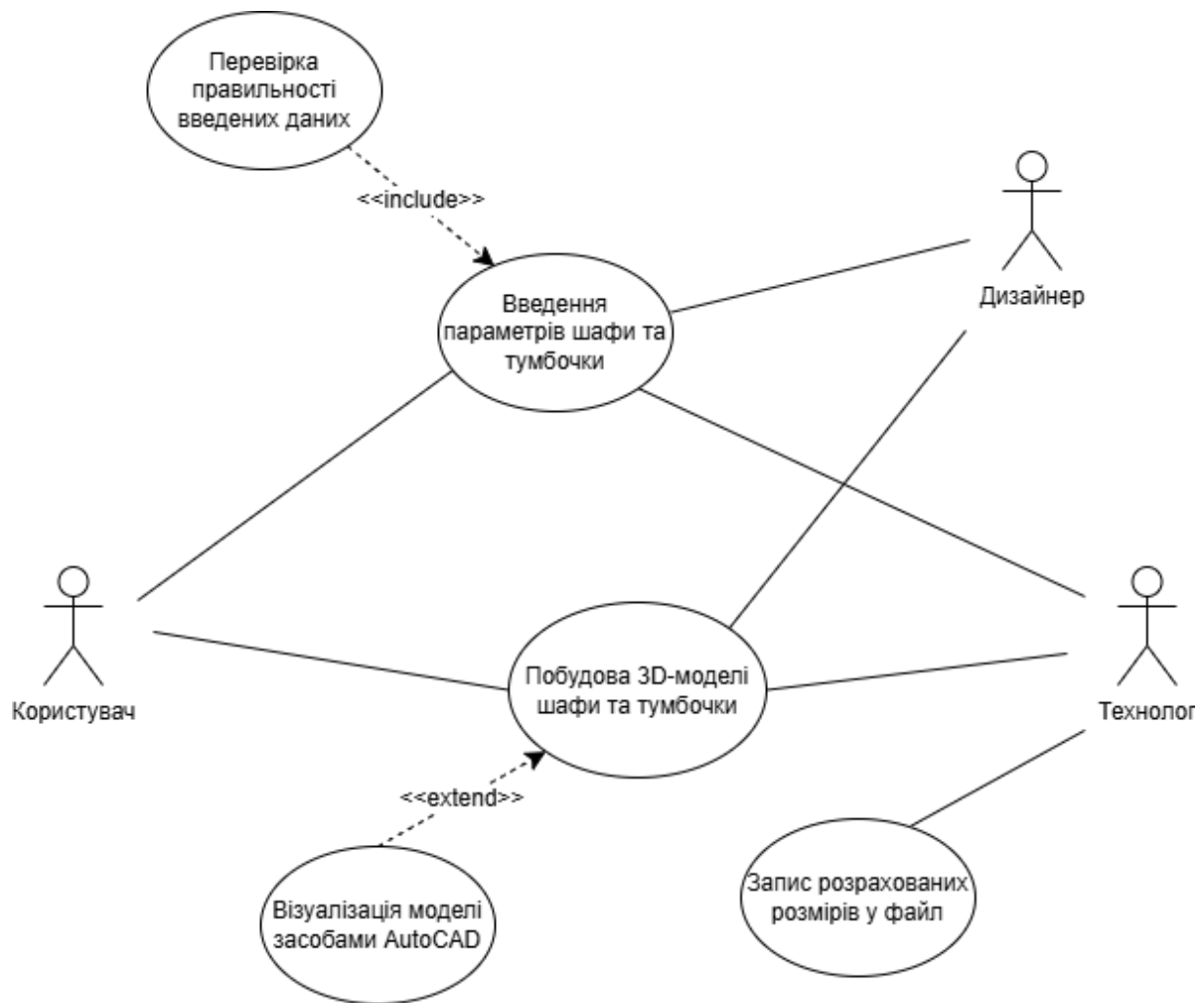


Рисунок 2.4 – Діаграма прецедентів проєктованої системи

## 2.5 Математичне та алгоритмічне забезпечення

У процесі створення системи автоматизованого проєктування мультифункціональної шафи було сформовано математичне та алгоритмічне забезпечення, що охоплює як обчислення всіх необхідних геометричних параметрів, так і логіку побудови моделі в середовищі AutoCAD за допомогою мови AutoLISP. Цей компонент системи є складовою частиною її концептуальної моделі, яка визначає зв'язки між вхідними даними, обчислюваними характеристиками та побудовою об'єктів у тривимірному просторі.

Математичне забезпечення полягає у формалізації всіх геометричних розрахунків, необхідних для побудови шафи й тумбочки. Основні вхідні параметри, які задає користувач через діалогове вікно (у тому числі ширина, висота, глибина,

товщина матеріалу, висота тумбочок, висота арки, параметри тумбочки), зчитуються як текстові значення, які далі перетворюються на числові за допомогою функції `atof`. Оскільки `atof` повертає значення `0.0` у випадку нечислового введення або порожнього рядка, цей механізм використовується як базовий засіб перевірки правильності введених даних: `(setq width (atof (get_tile "width"))) ...`).

Для уникнення помилок у геометричних розрахунках, у програмі реалізовано перевірку на коректність вхідних розмірів. Ці обмеження задаються через порівняння числових значень із мінімально допустимими межами, визначеними в межах інтерфейсу користувача:

`(if (or`

```
    (< width 400) (> width 2400)
    (< depth 300) (> depth 1000)
    (< height 800) (> height 2700)
    (< thickness 10) (> thickness 30)
    ...
  )
```

`(alert "Перевірте розміри. Ви ввели недопустимі значення або вийшли за межі допустимого значення.")`  
`)`.

Така перевірка забезпечує фільтрацію помилок ще до початку моделювання. Вона вбудована в обробник кнопки "ОК" у діалоговому вікні DCL через `action_tile` "асерт", тобто відбувається до закриття діалогу. Якщо значення некоректні, то замість завершення діалогу виводиться повідомлення (рис. 2.5), і користувач повертається до редагування: `(done_dialog 1)`. Логіка цієї перевірки представлена на блок-схемі (рис. 2.6): одразу після введення даних програма перевіряє чи це число, і чи воно в допустимих межах. Якщо умови не виконуються, відбувається блокування побудови та повторне введення. Це гарантує, що подальші етапи розрахунку або побудови 3D-моделі не будуть виконані на основі некоректних або відсутніх даних.

Якщо ж значення є коректним, алгоритм переходить до наступного етапу – формування моделі.

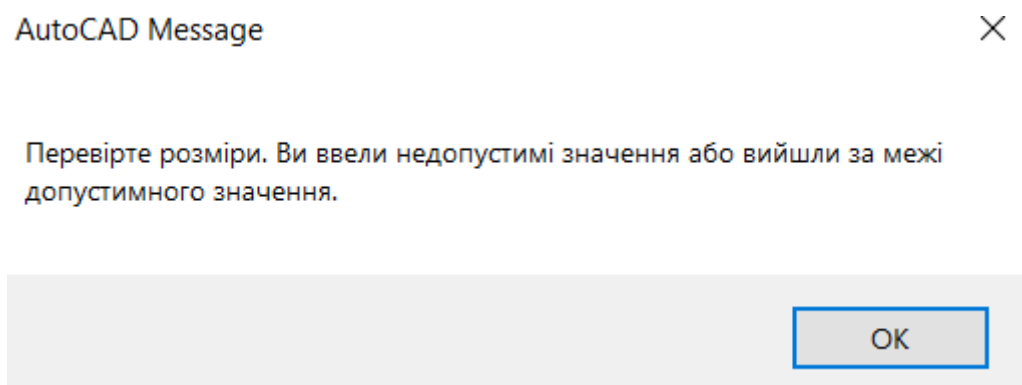


Рисунок 2.5 – Повідомлення про ввід недопустимого значення

Усі габарити деталей визначаються математично відповідно до вхідних параметрів, з урахуванням товщини матеріалу, кількості секцій і внутрішніх відступів.

Для кожного елемента можна умовно позначити параметри: загальна зовнішня ширина шафи , висота , глибина , внутрішні значення позначаються відповідно , , , товщина панелей – t тощо. За цим принципом для розрахунків використовуються наступні формули:

Для внутрішньої ширини шафи:

$$W_{\text{внут}} = W_{\text{заг}} - 2t \quad (2.1)$$

Для зовнішньої висоти шафи:

$$H_{\text{внут}} = H_{\text{заг}} - 2t \quad (2.2)$$

Для відстані між полицями:

$$\Delta h = \frac{H_{\text{внут}} - N * t}{N + 1} \quad (2.3)$$

Ширина кожної з двох симетричних дверцят вираховується за формулою:

$$W_{\text{дверц}} = \frac{W_{\text{внутр}}}{2} \quad (2.4)$$

Висота дверцят:

$$H_{\text{дверц}} = H_{\text{внутр}} \quad (2.5)$$

Центр арки:

$$x = \frac{W_{\text{внутр}}}{2}, \quad y = H_{\text{заг}} - \frac{t}{2} \quad (2.6)$$

Координати арки визначаються так, щоб її центр збігався з центром передньої верхньої площини шафи. Наприклад,  $x$ -координати центру арки дорівнює половині внутрішньої ширини. У кодї AutoLISP наведені вище формули реалізуються через присвоєння значень змінним. Наприклад, щоб задати внутрішню ширину, використовується така формула: (setq t3f (- width (\* 2 thickness))), тут width – це зовнішня ширина шафи, а thickness – це товщина боковин ( $t$ ), обчислюється як зовнішня ширина мінус дві товщини боковин[4].

Алгоритмічна частина забезпечення реалізує чітку послідовність дій: після зчитування і перевірки вхідних параметрів програма ініціалізує побудову моделі. Вона використовує функцію Active-X AddBox, яка створює прямокутний паралелепіпед, задаючи його початкову точку ( $x, y, z$ ) та габарити (довжина, ширина, висота) відповідно до результатів математичних обчислень у просторі AutoCAD. Кожна деталь – чи то боковина шафи, панель тумбочки чи дверцята – будується як паралелепіпед з урахуванням відповідних координат, отриманих з математичних розрахунків. Початкова точка моделі визначається користувачем при вставці:

```
(vla-AddBox model (vlax-3D-point (list cx cy cz)) dx dy dz).
```

Таким чином, математичне та алгоритмічне забезпечення в цій системі не лише формалізує просторові властивості меблів, а й гарантує, що побудова буде коректною, узгодженою з усіма обмеженнями і вимогами. Базова валідація значень на етапі введення мінімізує ймовірність помилок, підвищуючи загальну надійність і точність системи.

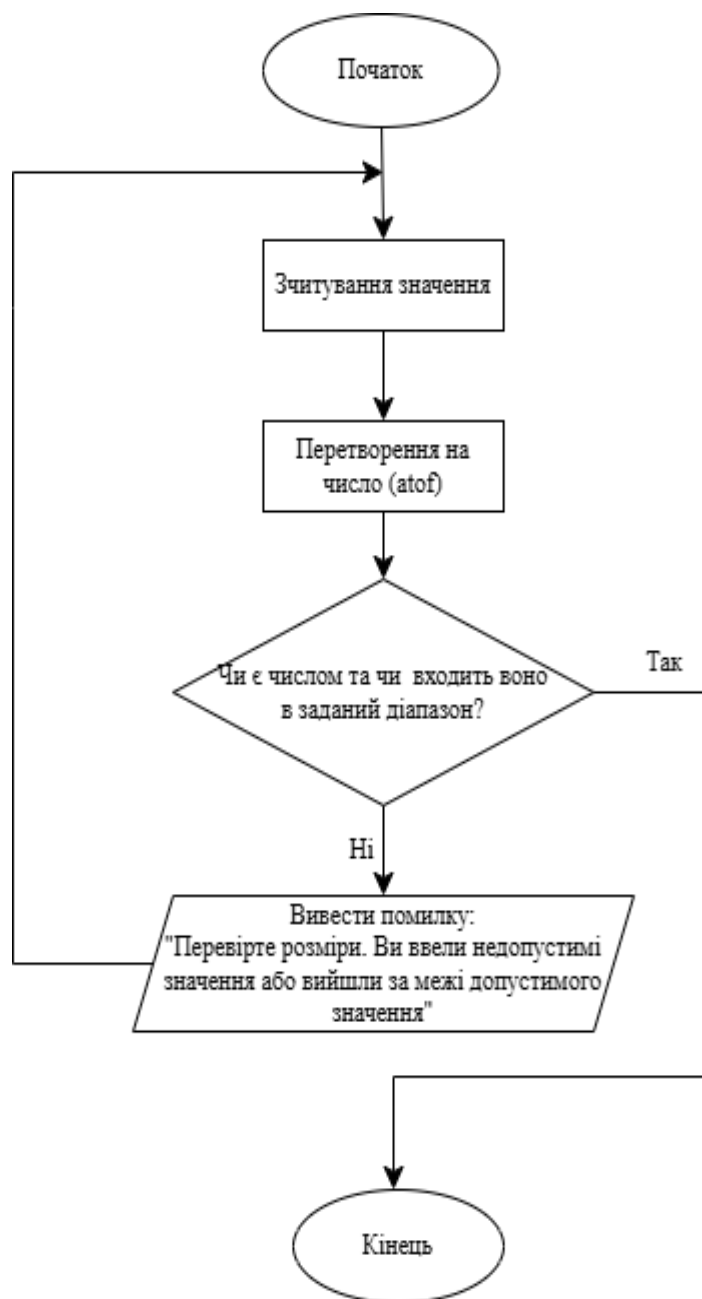


Рисунок 2.6 – Алгоритм роботи функції перевірки на число

У другому розділі було розроблено цілісне інформаційне, математичне та алгоритмічне забезпечення системи автоматизованого проєктування мультифункціональної шафи та тумбочки в середовищі AutoCAD. Проведений аналіз проблем ручного проєктування дозволив виокремити ключові труднощі - високу трудомісткість, неточність геометричних розрахунків, складність перевірки параметрів та обмеження стандартного функціоналу AutoCAD. Для візуалізації причинно-наслідкових зв'язків побудовано дерево проблем, яке трансформовано у

відповідне дерево цілей. Головна ціль – це створити інструмент, що автоматизує геометричні обчислення, перевірку параметрів і побудову тривимірної моделі меблів.

На основі цілей сформовано вимоги до програмного забезпечення. Описано функціональні блоки системи: введення параметрів, перевірка коректності даних, побудова 3D-моделі, генерація розрахункових специфікацій. Розроблена структурна схема системи відображає інформаційні потоки між модулями, зокрема, передачу параметрів, зворотний зв'язок та формування файлів документації. Система включає графічний інтерфейс на основі DCL, інтеграцію AutoLISP-алгоритмів, обробку даних у середовищі AutoCAD, що забезпечує стабільну та точну роботу.

Особливу увагу приділено математичному та алгоритмічному забезпеченню, у якому реалізовано перевірку типу введених параметрів, із відповідним повідомленням у разі помилки. Алгоритм перевірки гарантує, що всі значення є числовими та додатними, запобігаючи збоєм при обчисленнях. Усі геометричні розрахунки деталізовано через формули, які визначають розміри полиць, панелей, дверцят, арки. Побудова виконується за допомогою AutoLISP-функцій, що формують 3D-паралелепіпеди з точними координатами у просторі моделі.

Таким чином, розділ заклав концептуальну та технічну основу для надійного програмного рішення, яке підвищує точність, швидкість і надійність проектування корпусних меблів. Запропонована система автоматизує ключові етапи проектування, мінімізує ймовірність помилок і полегшує створення технічної документації.

## РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Основні функціональні можливості AutoCAD

AutoCAD – це широко відома система автоматизованого проектування, розроблена компанією Autodesk. Вона застосовується для створення двовимірних (2D) креслень і тривимірних (3D) моделей в різних галузях: від архітектури та будівництва до машинобудування і, зокрема, проектування меблів. AutoCAD надає користувачу потужний набір інструментів для точного конструювання геометричних об'єктів, виконання розрахунків, нанесення розмірів та створення технічної документації. Графічне середовище AutoCAD дозволяє будувати моделі з високою точністю, використовуючи систему координат, об'єкти-примітиви (лінії, дуги, кола, багатокутники тощо) та операції над ними (копіювання, переміщення, поворот, масштабування, відсікання та інші). Завдяки цьому проєктувальник може візуалізувати виріб до його фізичного виготовлення, перевірити відповідність розмірів, пропорцій і взаємного розташування деталей.

#### **Основні можливості AutoCAD охоплюють:**

##### 1. Створення двовимірних (2D) креслень:

- Побудова геометричних примітивів, з яких підтримуються лінії, кола, дуги, багатокутники, еліпси та інші базові фігури, що є основою для креслень.
- Передбачено команди для переміщення, копіювання, обертання, масштабування, обрізання, видовження та дзеркального відображення фігур.
- Можливість наносити лінійні, радіальні, кутові, діаметральні розміри, а також створювати багаторівневі текстові примітки з форматуванням.
- Система шарів дозволяє логічно структурувати креслення, вмикати/вимикати видимість, налаштовувати кольори, типи ліній, товщину тощо.
- Використання груп об'єктів - блоків для повторюваних об'єктів.

##### 2. Створення тривимірних (3D) моделей:

- Створення твердотільних, поверхневих та сітчастих об'єктів з параметрами довжини, висоти та ширини за допомогою різних команд.
  - Виконання операцій булевої геометрії (об'єднання, віднімання, перетин).
  - Застосування матеріалів і текстур для візуалізації реального вигляду.
  - Рендеринг об'єктів для створення фотореалістичних зображень моделей з різними ефектами (тінями, освітленням, матеріалами, відбиттями тощо).
3. Створення тривимірних (3D) моделей:
- Створення твердотільних, поверхневих та сітчастих об'єктів з параметрами довжини, висоти та ширини за допомогою різних команд.
  - Виконання операцій булевої геометрії (об'єднання, віднімання, перетин).
  - Застосування матеріалів і текстур для візуалізації реального вигляду.
  - Рендеринг об'єктів для створення фотореалістичних зображень моделей з різними ефектами (тінями, освітленням, матеріалами, відбиттями тощо).
4. Автоматизація проєктування:
- Використання макросів, скриптів і мікросценаріїв, що дозволяють автоматизувати лінійні дії за допомогою командного рядка або запису послідовності.
  - Підтримка AutoLISP (діалект мови Lisp), Visual LISP IDE, VBA, .NET API.
  - Розширення функціоналу AutoCAD під специфічні завдання користувача.
5. Сумісність та обмін даними:
- Підтримка форматів DWG, DXF, PDF тощо.
  - Інтеграція з іншими додатками Autodesk (Revit, Inventor, Civil 3D), а також експорт до САМ-систем (SolidCAM, Fusion 360).
  - Можливість спільної роботи через хмарні сервіси Autodesk, що дозволяє переглядати, коментувати та редагувати креслення онлайн, а також забезпечує синхронізацію файлів через Autodesk Drive або інші хмарні сервіси.

Важливою особливістю AutoCAD є підтримка тривимірного моделювання. Середовище надає інструменти для створення 3D-тіл (паралелепіпедів, циліндрів,

сфер), поверхонь та сітчастих моделей. Інженери можуть будувати деталі та складені вироби, виконувати булеві операції (об'єднання, перетин, віднімання), а також створювати креслення на основі 3D-моделей. У проєктуванні меблів це дозволяє створити реалістичну модель шафи й оцінити її конструкцію.

AutoCAD також має розвинені засоби автоматизації. Програма підтримує мову AutoLISP для створення власних функцій і модулів. Крім того, доступні Visual LISP, VBA, .NET API. Середовище Visual LISP IDE дозволяє писати й запускати код безпосередньо в AutoCAD. AutoLISP ідеально підходить для параметричного креслення, коли модель будується автоматично за заданими параметрами. У цьому проєкті саме AutoLISP використано для створення тривимірної моделі шафи, що робить AutoCAD ефективним інструментом для повної автоматизації процесу проєктування.

### **3.2 Обґрунтування вибору програмного забезпечення та інструментів**

Для розроблення системи автоматизованого проєктування мультифункціональної шафи необхідно обґрунтувати вибір програмної платформи та інструментів реалізації. Вибір було зроблено на користь AutoCAD з кількох причин. По-перше, AutoCAD є стандартизованим інструментом проєктування, який широко використовується у промисловості та навчанні. Використання цієї CAD-системи забезпечує сумісність результатів із загальноприйнятими форматами (DWG, DXF), що важливо для подальшого використання креслень і моделей у виробництві або обміні даними з колегами. По-друге, AutoCAD надає необхідну точність і функціональність для проєктування корпусних меблів: у програмі легко оперувати реальними розмірами, передбачені інструменти для побудови типових елементів конструкцій (прямокутників, отворів, отворів під фурнітуру тощо) та нанесення розмірів на кресленики.

Критичним фактором вибору AutoCAD стало вбудоване середовище програмування AutoLISP. AutoLISP було обрано як мову реалізації алгоритмів автоматичного побудування моделі шафи завдяки її тісній інтеграції з AutoCAD. На

відміну від зовнішніх застосунків, AutoLISP-програми виконуються безпосередньо в середині AutoCAD і мають доступ до команд креслення та об'єктів у поточному кресленні. Це дозволяє програмі безпосередньо створювати геометричні примітиви, модифікувати їх, запитувати в користувача необхідні дані через командний рядок або діалогові вікна. Іншими словами, AutoLISP забезпечує автоматизацію проектування у знайомому для інженера середовищі – немає потреби експортувати дані в інші системи чи використовувати зовнішні CAD-ядра. Окрім того, синтаксис AutoLISP досить простий для засвоєння, що прискорює розробку і відлагодження коду у порівнянні, наприклад, з використанням C++/ObjectARX чи C#.NET API, які вимагають глибшої програмної підготовки.

Інструментарій, використаний у розробці, включає середовище Visual LISP IDE, яке постачається разом з AutoCAD. У цьому середовищі було написано і відлагоджено програмний код на AutoLISP, а також створено файл діалогового інтерфейсу користувача мовою DCL. DCL було обрано для побудови графічного інтерфейсу, оскільки це стандартний засіб AutoCAD для опису діалогових вікон, що спрощує взаємодію користувача з програмою. Діалогові вікна дозволяють уникнути ручного введення параметрів через командний рядок, знижують ймовірність помилок і роблять програму більш зручною у використанні.

Таким чином, вибір AutoCAD як базової системи, AutoLISP як мови програмування, та DCL для створення інтерфейсу користувача є оптимальним для вирішення поставленого завдання. Ці інструменти забезпечують необхідну функціональність (3D-моделювання, точність креслення), гнучкість розробки та інтеграцію компонентів (інтерфейсу, алгоритмів та графічного середовища) в єдиному середовищі.

### **3.3 Реалізація програмного забезпечення**

У цьому підрозділі детально розглянуто процес розробки програмного модуля автоматизованого проектування шафи. Основою модуля є функція AutoLISP з ім'ям `s:connectedCabinet`, яка реалізує побудову моделі шафи за заданими параметрами.

Префікс *c:* в імені функції означає, що після завантаження файлу з кодом ця функція реєструється як користувацька команда AutoCAD. Таким чином, користувач може запустити побудову шафи, ввівши команду `CONNECTEDCABINET` у командному рядку AutoCAD (ім'я не чутливе до регістру) або викликавши її через інтерфейс (наприклад, призначивши на кнопку). Функція `c:connectedCabinet` містить реалізацію всіх необхідних кроків - від отримання вихідних даних до безпосереднього створення тривимірних об'єктів та запису результатів. Логіку роботи цієї функції можна описати поетапно:

### **1. Ініціалізація та введення даних:**

При запуску функції відбувається ініціалізація необхідних змінних та запуск діалогового вікна для введення параметрів. Користувач вводить основні габаритні розміри шафи та інші параметри (висоти, ширини секцій, глибину, товщину матеріалу, кількість полиць тощо). Після підтвердження введення (натиснення кнопки "ОК" в діалозі) програма зчитує ці значення, перевіряє їх коректність і переходить до наступного кроку.

### **2. Обробка параметрів і обчислення координат:**

Зі зчитаних параметрів обчислюються допоміжні значення, необхідні для побудови геометрії. Наприклад, розраховується внутрішня ширина секцій з урахуванням товщини матеріалу (щоб полиці правильно вписалися між боковинами), висота розташування полиць (якщо їх кілька, можуть обчислюватися координати кожної полиці з рівномірним або заданим кроком), висота аркового елемента тощо. На цьому етапі програма створює локальні змінні для збереження кожного параметра та похідного значення – це підвищує читабельність коду і спрощує внесення змін.

У випадку розміщення двох внутрішніх полиць у шафі простір між тумбочкою (висота  $h_1$ ) і верхньою панеллю (`bodyHeight`) поділяється на три рівномірні інтервали. Це потрібно для того, щоб полиці були рівномірно розташовані по висоті, з однаковими проміжками між ними.

Розрахунок інтервалу (`stepHeight`) виконується за формулою:

$$\text{stepHeight} = \frac{\text{bodyHeight} - h_1 - t}{3} \quad (3.1)$$

Еквівалент у кодї має вигляд: (setq stepHeight (/ (- bodyHeight h1 thickness) 3.0))

Вона вираховує висоту тіла шафи bodyHeight, висоту нижнього блоку h1, а також товщину полиці thickness. Різниця між цими значеннями визначає простір, доступний для розміщення полицок. Отже, перша полицка розміщується на висоті , а друга – на висоті . Таким чином, весь внутрішній простір між тумбочкою та верхньою панеллю поділяється на 3 рівні частини для рівномірного налаштування 2 полиць:

```
(setq t8sf (list (+ x thickness) (+ y thickness) (+ z (+ h1 stepHeight))))
```

```
(setq t9sf (list (+ x thickness) (+ y thickness) (+ z (+ h1 (* 2 stepHeight)))))
```

### 3. Побудова каркасу шафи:

Наступним кроком є безпосереднє створення об'єктів, що відповідають деталям шафи. Програма починає з базових елементів каркасу - боковини, горизонтальні панелі, тощо. Наприклад, створюються бічні стінки шафи: ліва і права панелі. Для цього задаються прямокутні контури розміром "висота x глибина", які потім перетворюються у тривимірні тверді тіла товщиною, рівною товщині матеріалу. Створення прямокутника та його екструзія здійснюється засобами AutoLISP через виклик відповідних команд AutoCAD. У разі боковин: довжина = висота шафи, ширина = товщина матеріалу, висота (в команді BOX) = глибина шафи. Таким чином, боковина виходить як вертикальний прямокутний паралелепіпед потрібних розмірів, розташований у просторі моделі. За такою ж логікою створюється також тумбочка.

Приклад створення задньої панелі шафи:

```
(setq t3sf (list (+ x thickness) (+ y (- depth thickness)) z))
```

```
(Addbox t3sf (- width (* 2 thickness)) thickness bodyHeight)
```

```
(setq b_backPanel (list bodyHeight (- width (* 2 thickness)))))
```

Де t3sf – це точка вставки (тобто ліва нижня передня точка) задньої панелі шафи, (+ x thickness) - це зміщення по осі X, щоб уникнути перекриття з лівою боковою панеллю, (+ y (- depth thickness)) - позиція по осі Y, яка визначає задню площину шафи. Тобто ми рухаємося на глибину (depth), але віднімаємо товщину, бо задня панель має бути не зовні, а всередині конструкції, а z – початкова координата по Z, тобто нижня межа панелі по вертикалі. Функція AddBox створює 3D-прямокутник беручи повну ширину шафи без урахування лівої та правої панелі, та повну висоту

шафи. Setq створює список b\_backPanel, який містить розміри задньої панелі (висоту та ширину). Він потрібен для запису виводу у файл, щоб зберегти габарити деталі.

#### **4. Побудова внутрішніх полиць і перегородок:**

Після створення боковин, функція c:connectedCabinet генерує полиці та інші внутрішні елементи (наприклад, перегородка між полицями, яка присутня в шафі). Полиці являють собою горизонтальні панелі. Для кожної полиці обчислюється її розташування: координати переднього лівого кута на основі габаритів шафи і заданої висоти полиці. Полиці будуються як прямокутні паралелепіпеди з розмірами "глибина x ширина секції " без врахування товщини. Наприклад, якщо шафа має внутрішню ширину 800 мм між боковинами і глибину 500 мм, то полиця буде 800 x 500 мм.

#### **5. Побудова арочного елемента:**

Особливістю розробленої системи автоматизованого проєктування є наявність арочного верхнього елемента в одній із секцій шафи.

Для створення арки застосовано поетапне формування за допомогою полілінії з дугою і подальшої екструзії (Додаток А). Спочатку будується зовнішній контур арки (полілінія з дуговим сегментом), далі цей контур екструдовано на задану глибину шафи. Потім за аналогічною схемою створюється внутрішній контур арки, зміщений на товщину матеріалу всередину. Врешті, за допомогою булевої операції SUBTRACT внутрішній контур вирізається із зовнішнього, утворюючи порожнистий арковий елемент з відповідною товщиною стінки.

- (vl-cmdf "\_UCS" "\_Front") – цей рядок встановлює систему координат UCS у фронтальну площину, щоб побудова арки відбувалась у вертикальній орієнтації (по осях X і Z, глибина – по Y).
- Потім визначаються координати нижнього лівого кута арки в новій UCS. Значення z1 визначається як від'ємна глибина шафи, оскільки фронтальна площина дивиться ззаду.

- Створюється замкнута полілінія у форму арки, що складається з вертикального відрізка зліва ( $t1o$  до  $t3o$ ), дуги (ключ "\_A") між верхніми точками, вертикального відрізка справа ( $t4o$  до  $t2o$ ), команда "\_C" замикає контур [5].
- Потім екструдується і створюється зовнішній твердий об'єкт арки, аналогічно після цього будується внутрішня арка, але всі координати зсунуті всередину на товщину *thickness*. Після екструзії утворюється внутрішній твердий об'єм, який віднімається від зовнішнього. В результаті формується аркова стінка заданої товщини.

#### **6. Побудова з'єднаного модуля шафи:**

Мультифункціональна шафа складається не тільки з основного високого відділення, але й з додаткової тумбочки, з'єднаної з основною шафою (рис. 3.1). Функція *s:connectedCabinet* продовжує роботу, додаючи геометрію цього модуля. Для додаткової тумбочки використовуються ті самі принципи: боковини, дно, верх та внутрішня поличка. Оскільки полиця сполучена з шафою, програма повинна коректно сумістити їх геометрично. Програма враховує це: після будівництва шафи будується і суміжна стінка, що може бути незалежною за потреби. Всі деталі теж створюються як паралелепіпеди заданих розмірів і розташовуються за координатами, розрахованими на основі параметрів другої секції (її висоти, ширини, глибини). Завдяки тому, що всі координати і розміри прив'язані до єдиної системи координат AutoCAD, обидві секції будуються як єдина тривимірна модель, всі елементи якої узгоджені між собою.

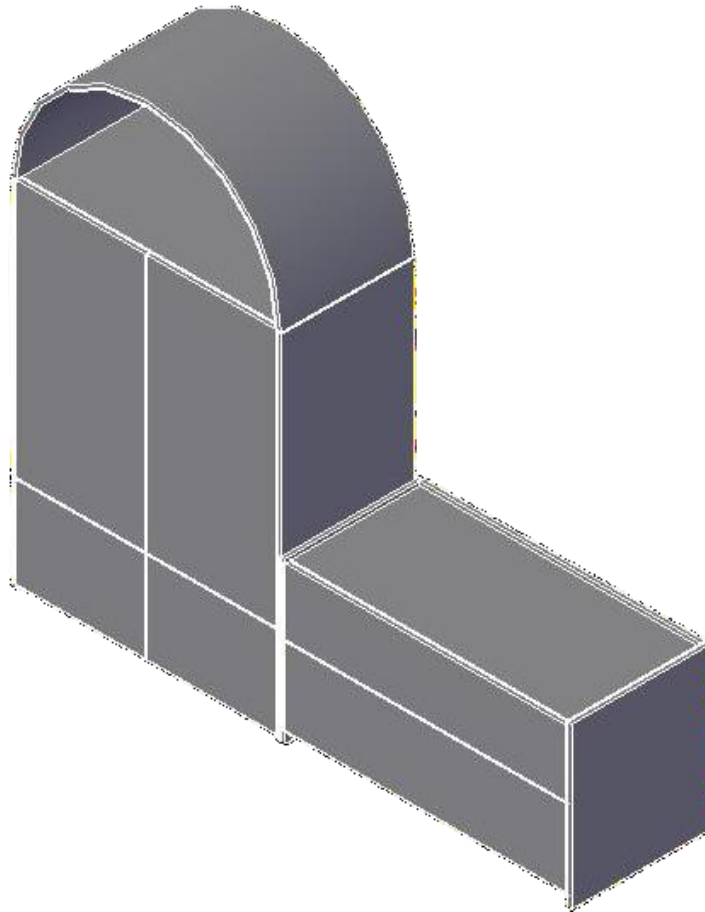


Рисунок 3.1 - Концептуальний вигляд шафи у середовищі AutoCAD 2025

## **7. Виведення результатів:**

Завершивши геометричне моделювання, функція `s:connectedCabinet` формує підсумкові результати. Користувач може візуально перевірити конструкцію, використовуючи засоби візуалізації AutoCAD (наприклад, обертання моделі, перерізи тощо).

Більше того, програма генерує текстовий звіт з розмірами деталей, про що детально йдеться у підрозділі 3.5. Цей звіт потрібен для подальшого виготовлення шафи і містить перелік усіх панелей із зазначенням їх габаритних розмірів та інших характеристик (наприклад, інформації про обробку кромки). Після успішного завершення роботи функції на екрані або в командному рядку може відобразитися повідомлення про те, що модель побудовано і файл результатів збережено – це слугує зворотним зв'язком для користувача.

Також для наглядності була розроблена блок-схема роботи алгоритму функції `s:connectedCabinet` (рис 3.2), яка передає всю суть функції.

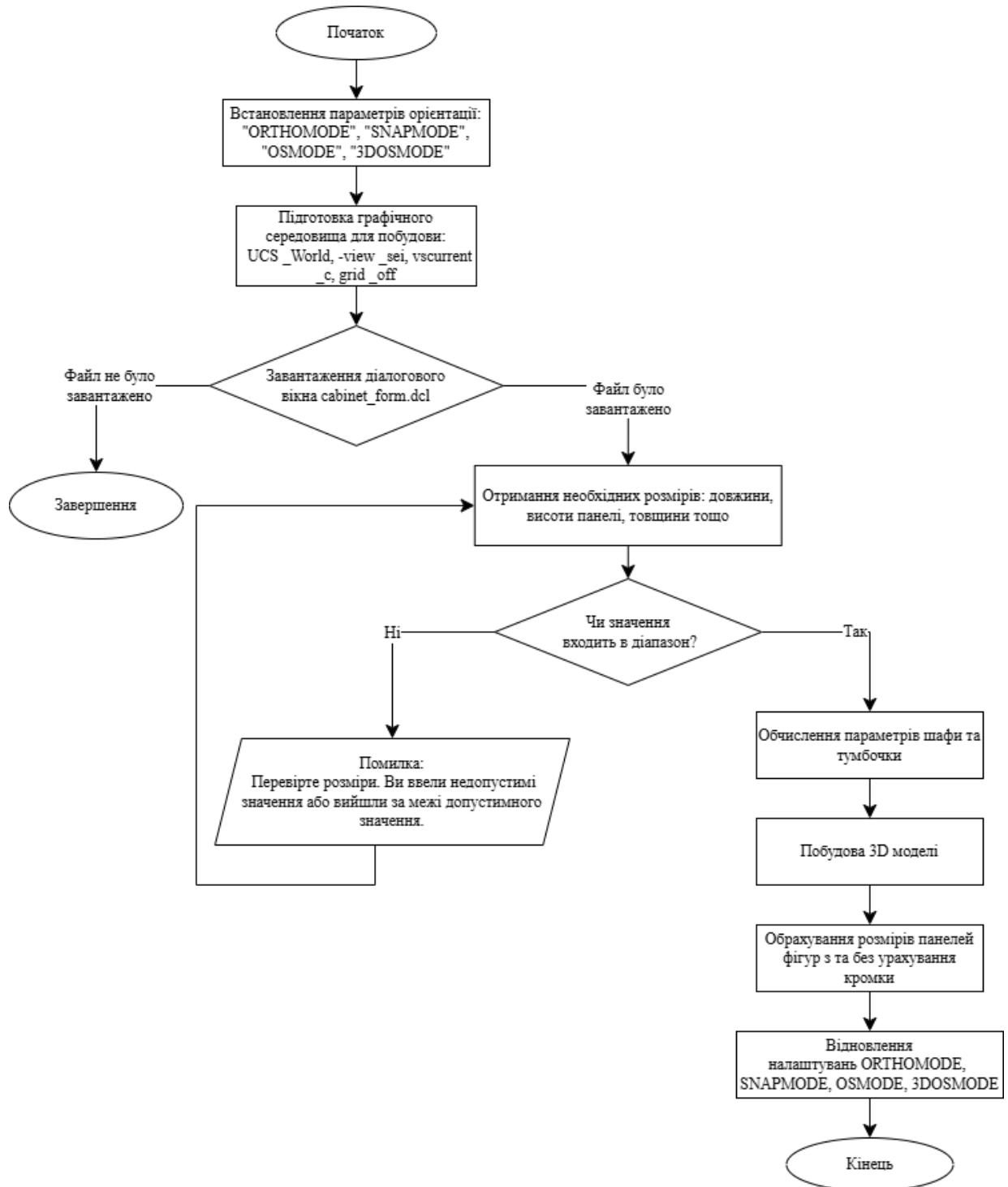


Рисунок 3.2 – Алгоритм роботи функції `connectedCabinet`

На початковому етапі відбувається встановлення параметрів середовища (ORTHOMODE, SNAPMODE, OSMODE, 3DOSMODE) та підготовка простору

моделювання: задається світова система координат, змінюється вид, відключається сітка. Далі система намагається завантажити діалогове вікно, і якщо файл .dcl не знайдено, виконання завершується.

Після успішного завантаження форми користувач вводить вхідні параметри, які зчитуються у вигляді текстових значень. Значення перевіряються функцією `atof`, яка конвертує текст у числа. На основі цього реалізовано механізм перевірки валідності введення - програма одразу визначає, чи всі входять вони у допустимий діапазон. У разі некоректного введення система повідомляє про помилку й повертається до введення значень, не переходячи до наступних етапів.

Якщо всі дані коректні, відбувається обчислення розмірів елементів шафи й тумбочки, після чого здійснюється побудова 3D-моделі у просторі AutoCAD за допомогою ActiveX-функції `AddBox`. Сформовані параметри використовуються також для розрахунку заготовок - як з урахуванням личківки, так і без неї. Після завершення побудови, система повертає попередні значення системних змінних, щоб не впливати на подальшу роботу користувача в середовищі AutoCAD.

### **3.4 Інтерфейс користувача та взаємодія**

Ефективна робота користувача з програмою автоматизованого проектування значною мірою залежить від зручності інтерфейсу. У розробленому модулі вирішено використати графічний інтерфейс на основі діалогового вікна, реалізованого засобами DCL (рис. 3.3). На відміну від стандартного введення параметрів через командний рядок, діалогове вікно дозволяє відобразити всі потрібні поля вводу, підписи до них, пояснення одиниць виміру, а також забезпечити базову перевірку правильності даних ще до побудови моделі.

Додаток AutoCAD для побудови шафи та тумбочки ×

Шафа	
Ширина:	1220
Глибина:	600
Висота:	1610
Нижня полиця:	420
Товщина:	18

Тумбочка	
Ширина:	1320
Глибина:	630
Висота:	710
Верхня полиця:	300
Середня полиця:	410
Нижня полиця:	70
Товщина:	18

Налаштування	
Товщина кромки:	1 мм
<input checked="" type="checkbox"/> Записати дані у файл	

OK
Скасувати

Рисунок 3.3– Діалогове вікно побудови шафи

Діалогове вікно створено з використанням DCL-файлу (рис 3.4), в якому описано макет і логіку взаємодії. Основним діалогом є cabinetDialog. У ньому згруповано параметри для основної шафи та додаткової тумбочки у два окремі блоки за допомогою :boxed\_column. Для кожного параметра передбачено текстову мітку та поле вводу (edit\_box), вирівняні по горизонталі за допомогою :row. Така структура робить форму читабельною та зручною у використанні. Заголовок діалогу чітко вказує на його призначення, наприклад: "Параметри шафи та тумбочки".

```
cabinet_form.DCL
cabinetDialog : dialog {
  label = "Додаток AutoCAD для побудови шафи та тумбочки";
  : column {
    : boxed_column {
      label = "Шафа";
      : row { : text { label = "Ширина:"; alignment = left; } : edit_box { key = "width"; edit_width = 30; } }
      : row { : text { label = "Глибина:"; alignment = left; } : edit_box { key = "depth"; edit_width = 30; } }
      : row { : text { label = "Висота:"; alignment = left; } : edit_box { key = "height"; edit_width = 30; } }
      : row { : text { label = "Нижня полиця:"; alignment = left; } : edit_box { key = "h1"; edit_width = 30; } }
      : row { : text { label = "Висота арки:"; alignment = left; } : edit_box { key = "archHeight"; edit_width = 30; } }
      : row { : text { label = "Товщина:"; alignment = left; } : edit_box { key = "thickness"; edit_width = 30; } }
    }
    : boxed_column {
      label = "Тумбочка";
      : row { : text { label = "Ширина:"; alignment = left; } : edit_box { key = "width_tb"; edit_width = 30; } }
      : row { : text { label = "Глибина:"; alignment = left; } : edit_box { key = "depth_tb"; edit_width = 30; } }
      : row { : text { label = "Висота:"; alignment = left; } : edit_box { key = "height_tb"; edit_width = 30; } }
      : row { : text { label = "Верхня полиця:"; alignment = left; } : edit_box { key = "h_verh"; edit_width = 30; } }
      : row { : text { label = "Середня полиця:"; alignment = left; } : edit_box { key = "h_nyz"; edit_width = 30; } }
      : row { : text { label = "Нижня полиця:"; alignment = left; } : edit_box { key = "pid_vysota"; edit_width = 30; } }
      : row { : text { label = "Товщина:"; alignment = left; } : edit_box { key = "thickness_tb"; edit_width = 30; } }
    }
  }
  : row {
    : button { label = "ОК"; key = "accept"; is_default = true; }
    : button { label = "Скасувати"; key = "cancel"; is_cancel = true; }
  }
}
}
```

Рисунок 3.4 – Програмний код діалогового вікна на мові DCL

У діалозі реалізовано введення розмірів основної шафи: висота, ширина, глибина, висота нижньої полиці, висота арки, товщина матеріалу, введення параметрів тумбочки: ширина, глибина, висота, висота верхньої та середньої полиць, висота ніжок, товщина плити, самі поля мають підписи з поясненням одиниць виміру (мм) та заповнюються значеннями за замовчуванням через (get\_tile ...). Також є кнопки "ОК" (accept) і "Скасувати" (cancel), які дозволяють користувачу підтвердити введені значення або відмовитися від дій.

У коді AutoLISP використано load\_dialog, new\_dialog для ініціалізації, set\_tile для заповнення значень, action\_tile для обробки подій. Коли користувач натискає "ОК", усі значення зчитуються через (get\_tile ...) і перетворюються у числа за допомогою (atof ...). У разі помилок (вихід за межі допустимого значення) виводиться alert із повідомленням. Якщо ж дані коректні, діалог закривається (done\_dialog 1) і передається керування основному коду.

Після цього користувачеві пропонується вказати точку вставки шафи (рис 3.5) на кресленні, це реалізовано через функцію (getpoint ...) [6].

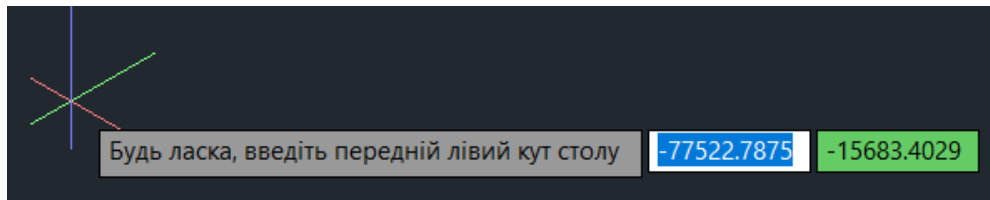


Рисунок 3.5 – Повідомлення для вибору початкової точки вставки шафи

Такий підхід забезпечує повну інтеграцію графічного інтерфейсу з логікою побудови моделі. Інтерфейс не допускає порожніх або некоректних значень. Завдяки тому, що кожне поле має унікальний key, пов'язаний із змінною у LISP-кодi, параметри легко зчитуються, обробляються і використовуються при побудові.

Таким чином, розроблений інтерфейс на базі DCL забезпечує просту, надійну та масштабовану взаємодію користувача з програмою, що робить процес введення параметрів зручним і знижує ймовірність помилок при побудові моделі.

### 3.5 Формування результатів

Одним із результатів роботи програми є автоматичне формування специфікації деталей – тобто створення файлу, де перелічені всі складові елементи шафи з їх розмірами (рис. 3.6). Такий файл є надзвичайно корисним на етапі виготовлення меблів, оскільки фактично являє собою карту розкрою або принаймні список заготовок, які необхідно вирізати з плитного матеріалу.

Програма `s:connectedCabinet` після успішного побудування 3D-моделі виконує процедуру запису даних про деталі. Ця процедура проходить наступні кроки:

- **Збирання інформації про деталі.** Протягом побудови моделі програма знає параметри кожної деталі. На етапі формування специфікації всі ці параметри структуруються. Для кожної деталі визначається її назва та розміри. Розміри фіксуються у вигляді довжина x ширина для того аби врахувати напрям волокон при реальній побудові моделей. Тут під довжиною мається на увазі найбільший розмір деталі (наприклад, для боковини – це висота, для полиці – ширина між боковинами), ширина – другий за величиною розмір (для боковини – глибина, для полиці – глибина шафи).

- **Підготовка файлу.** AutoLISP дозволяє працювати з файлами через функції вводу-виводу. Програма відкриває файл для запису, використовуючи (open filename "w"), де filename - рядок, що містить шлях і назву файлу. Назву файлу було визначено наперед (cabinet\_output.txt).
- **Запис даних у файл.** Програма записує інформацію про кожну деталь. Кожен рядок містить назву деталі та її розміри (рис 3.6). Формат представлення вийшов таким: <Назва деталі>: (<довжина> <ширина> ). Реальні значення залежать від введених користувачем параметрів.

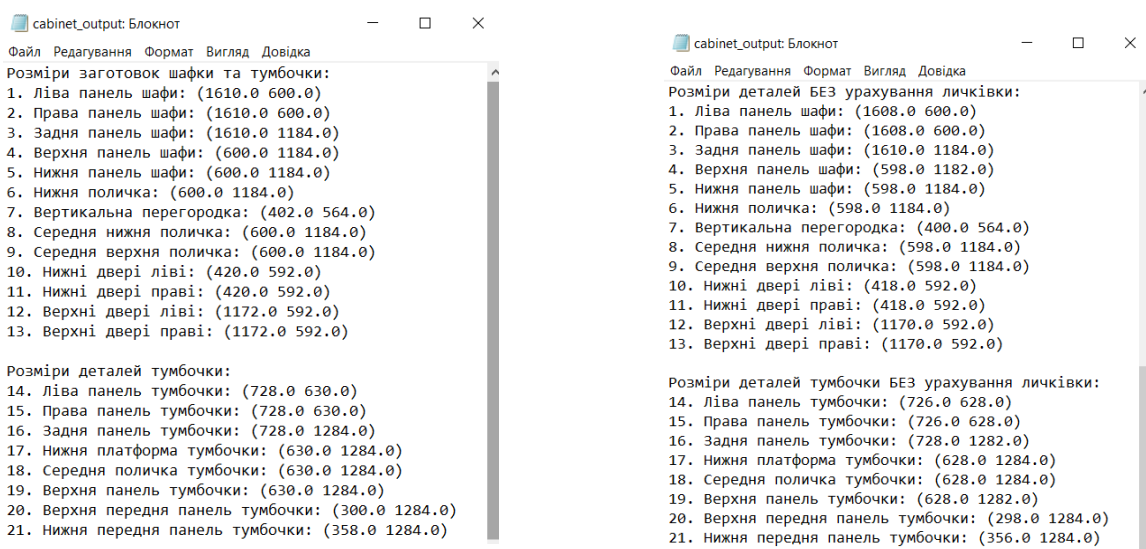


Рисунок 3.6 – Результат запису у файл

Файл зі списком деталей виступає як зв'язуюча ланка між етапом проектування і виробництвом. Інформація з нього може бути використана технологом або майстром для планування розкрою матеріалу. Крім того, на основі цих даних можна оцінити кількість матеріалу, необхідного для виготовлення шафи, що важливо для економічних розрахунків.

Важливо, що специфікація, створена програмою, повністю синхронізована з побудованою 3D-моделлю. Це означає, що виключається ризик розбіжностей між кресленням і списком деталей, які часто трапляються при ручному підрахунку. Кожен параметр, введений користувачем, впливає і на модель, і на вихідний файл, тому при змінах (наприклад, користувач вирішив збільшити висоту) досить повторно

запустити команду з новими даними, і буде автоматично отримано оновлену специфікацію.

### **3.6 Особливості реалізації обробки кромки у деталях шафи та тумбочки**

Обробка кромки деталей меблів є важливою складовою етапу підготовки виробу до виготовлення. У випадку шафи та тумбочки, які моделюються в рамках цього проєкту, кромкування дозволяє не лише покращити зовнішній вигляд виробу, а й захистити краї плитних матеріалів від механічних пошкоджень та впливу вологи. У межах даного програмного рішення AutoLISP-реалізації було створено функціонал, який враховує кромки як під час побудови моделі, так і в процесі автоматичного формування специфікації деталей.

У програмному коді реалізовано спеціальний блок функцій для обробки кромки, які відповідають за обчислення розмірів деталей з урахуванням і без урахування личківки. Це функції `obrizka1`, `obrizka_kromku1`, `obrizka_kromku2`, `obrizka_kromku12` (повна реалізація функцій є в Додатку А). Вони працюють за наступним принципом:

- `obrizka1` – базова функція, яка повертає абсолютні розміри без змін.
- `obrizka_kromku1` – модифікує довжину деталі, імітуючи зменшення за рахунок кромкування з фронтальної сторони.
- `obrizka_kromku2` – модифікує другу координату списку розмірів, яка в залежності від типу деталі може відповідати ширині або висоті. Вона зменшується на товщину кромки, якщо обробляється верхній або нижній край деталі.
- `obrizka_kromku12` – враховує наявність кромки одночасно з обох боків: і фронтальної, і верхньої.
- `obrizka_kromkuAll` – модифікує деталь, враховуючи всі чотири боки: верхньої, нижньої, правої та лівої.

Змінна `kromka` задає товщину кромки, яка використовується в обчисленнях. Присутня також можливість вибору товщини кромки – 0.4, 0.8, 1.0, 2.0, 3.0 мм.

Для кожної деталі, побудованої функцією AddBox, зберігається відповідна змінна (наприклад, b\_top, b\_shelf1, tb\_middleShelf тощо), яка містить список габаритів. Після побудови моделі програма виконує запис у текстовий файл, де виводиться специфікація деталей у двох варіантах:

1. З урахуванням повних розмірів.
2. Без урахування личківки - розміри зменшені відповідно до сторін, які підлягають кромкуванню.

У коді передбачено логіку, яка для кожної деталі вибирає потрібну функцію обробки, залежно від її розташування у виробі та вимог до кромкування. Наприклад:

- Боковини шафи кромкуються з переднього торця (obrizka\_kromku1).
- Верхня панель – з двох сторін (obrizka\_kromku12).
- Полиці – лише по передній кромці (obrizka\_kromku1).
- Тильна панель шафи (b\_backPanel) – не має кромки, і тому використовується базова obrizka1.
- Двері – мають обробку зі всіх сторін (obrizka\_kromkuAll).

Завдяки такій реалізації, програма автоматично генерує правильні розміри деталей для виробництва, що знижує ймовірність помилок та економить час на підготовку проектної документації. Обробка кромки не моделюється візуально в AutoCAD, проте вся необхідна інформація про неї надається у вихідному текстовому файлі, що відповідає вимогам меблевого виробництва.

Таким чином, розроблене рішення дозволяє інтегрувати облік технологічної операції кромкування в загальний цикл автоматизованого проектування меблів, забезпечуючи повноцінний вихідний набір даних для подальшого виробничого процесу.

У третьому розділі здійснено практичну реалізацію системи автоматизованого проектування мультифункціональної шафи засобами AutoCAD. Було обґрунтовано вибір платформи AutoCAD та мови програмування AutoLISP для вирішення поставленого завдання, зважаючи на їхні можливості в 3D-моделюванні та автоматизації. Розроблено ПЗ у вигляді AutoLISP-функції c:connectedCabinet, яка

поєднує в собі всі етапи – від введення параметрів виробу до побудови тривимірної моделі і формування вихідних даних.

У підрозділах розкрито структуру та логіку роботи програми: детально описано алгоритм створення моделі шафи, включно з геометричними розрахунками, побудовою основних конструктивних елементів (корпусу шафи, полиць, додаткової секції-тумбочки, арочного верху). Окрему увагу приділено розробленню інтерфейсу користувача на основі діалогового вікна DCL, що забезпечило зручне введення вихідних даних і інтерактивну взаємодію. Описано, як програма формує підсумковий файл зі списком деталей та їх розмірами, що є необхідним для виробничого етапу (розкрою матеріалів). Впроваджено обробку кромки: визначено, які торці деталей потребують кромкування, і цю інформацію включено у результати проектування. Також реалізовано перевірку коректності введених даних – програма контролює, щоб користувач вводив лише допустимі числові значення параметрів, запобігаючи некоректним ситуаціям і помилкам.

Таким чином, розроблена система відповідає поставленим вимогам: вона дозволяє швидко і безпомилково отримувати параметричну 3D-модель мультифункціональної шафи за заданими розмірами та візуалізувати її (рис 3.7), автоматично генерує допоміжні дані для виготовлення та враховує технологічні нюанси.



Рисунок 3.7 – Візуалізований вигляд шафи та тумбочки

Розроблена шафа є мультифункціональною, оскільки поєднує в собі одразу кілька функціональних блоків, які зазвичай реалізуються окремо. Зокрема, ліва конструкція виконує роль класичної шафи для зберігання одягу або аксесуарів, у той час як права – тумбочка, що забезпечує робочу або відпочинкову поверхню й містить секції з шухлядами для організації простору.

Додатковим функціональним елементом виступає декоративна арка, яка може бути використана як для підсвічування, декору так і для зберігання.

Мультифункціональність також проявляється в можливості адаптації розмірів: система дозволяє користувачу самостійно змінювати параметри (висоту, ширину, глибину, кількість секцій тощо), що дає змогу застосовувати конструкцію в різних інтер'єрних сценаріях — від віталень і спалень до офісних приміщень.

## ВИСНОВКИ

У цій дипломній роботі було розроблено систему автоматизованого проектування мультифункціональної шафи в середовищі AutoCAD із використанням AutoLISP, Visual LISP та DCL. Такий підхід забезпечив швидке створення 3D-моделі виробу на основі параметрів користувача та високу точність побудови.

На першому етапі дослідження було проаналізовано особливості сучасного меблевого проектування, а також проведено порівняльний огляд популярних CAD-систем. Встановлено, що AutoCAD, завдяки своїм засобам автоматизації (AutoLISP, DCL), є оптимальним інструментом для створення індивідуальних програмних рішень у сфері меблевого дизайну.

У другому розділі сформовано повну інформаційну, математичну та алгоритмічну модель проекрованої системи. Побудовано дерево проблем і цілей, що дозволило логічно структурувати процес автоматизації. Реалізовано точні розрахунки геометричних параметрів деталей і алгоритми перевірки коректності введених значень, що забезпечує надійність і стабільність роботи програми.

У третьому розділі виконано безпосередню реалізацію системи: створено AutoLISP-функцію `s:connectedCabinet`, яка покриває повний цикл побудови виробу - від зчитування параметрів і побудови 3D-моделі до формування списку деталей у вигляді текстового файлу. Забезпечено обробку кромки, коректне розміщення деталей, генерацію двох варіантів специфікації (з кромкою і без). Розроблений графічний інтерфейс (DCL) робить систему інтуїтивно зрозумілою та зручною для користувача. Також реалізовано перевірку правильності введених значень, що запобігає помилкам.

Таким чином, розроблене програмне забезпечення повністю відповідає поставленій меті: дозволяє значно скоротити час проектування корпусних меблів, мінімізувати людський фактор у розрахунках, підвищити точність креслень та забезпечити повну підготовку до виготовлення виробу. Система є масштабованою й може бути адаптована до інших типів меблів, що відкриває перспективи подальшого розвитку і впровадження у виробничі процеси малих та середніх підприємств.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ambrosius L. Creating and Modifying Drawing Objects. AutoCAD Platform Customization / L. Ambrosius. Autodesk : Sybex, 2015. С. 83–112.
2. Molzahn P. A., Umstattd W. D., Davis C. W. Section 2 - Design and Layot. Modern Cabinetmaking / P. A. Molzahn, W. D. Umstattd, C. W. Davis. 6-те вид. Tinley Park, Illinois : The Goodheart-Willcox, 2023. С. 61–180.
3. Дячук З. Конструювання меблів. Столи, стільці та крісла меблі для відпочинку взаємозамінність, міцність. Київ : Києво-Могилянська академія, 2011. 484 с.
4. Omura G. The ABC's of AutoLISP. San Francisco : Sybex. 366 с.
5. Autodesk. AutoLISP Developer's Guide (AutoLISP). Autodesk. Autodesk AutoCAD 2025. URL: <https://help.autodesk.com/view/OARX/2025/ENU/> (дата звернення: 13.04.2025).
6. Kramer B., Kramer W., Kramer D. Understanding AutoLISP: Programming for Productivity. Albany, New York : Delmar, 1993. 225 с.
7. Ramage K. AutoLISP Beginners' Tutorials. AfraLISP. AfraLISP Learn AutoLISP for AutoCAD productivity. URL: [https://www.afralisp.net/autolisp/tutorials/index.php?category\\_id=1](https://www.afralisp.net/autolisp/tutorials/index.php?category_id=1) (дата звернення: 18.04.2025).
8. Mayfield F. AutoLISP to Automate Your Tasks: Part 2 – Intermediate. Autodesk. AutoCAD Blog. 06.10.2020. URL: <https://www.autodesk.com/blogs/autocad/autolisp-to-automate-your-tasks-part-2-intermediate/> (дата звернення: 23.04.2025).
9. Kumari C., Rana D. P. The Shaping of Spaces: A Journey Through Furniture Design History. ResearchGate. International Journal of Housing and Human Settlement Planning. 25.10.2023. URL: [https://www.researchgate.net/publication/380854583\\_International\\_Journal\\_of\\_Housing\\_and\\_Human\\_Settlement\\_Planning\\_The\\_Shaping\\_of\\_Spaces\\_A\\_Journey\\_Through\\_Furniture\\_Design\\_History](https://www.researchgate.net/publication/380854583_International_Journal_of_Housing_and_Human_Settlement_Planning_The_Shaping_of_Spaces_A_Journey_Through_Furniture_Design_History) (дата звернення: 25.04.2025).
10. Мигаль С. Проектування Меблів. Львів : Світ, 1999. 216 с.

11. Prabhu. AutoLISP Programming in AutoCAD. YouTube. cyber CAD solutions.  
URL: [https://www.youtube.com/playlist?list=PL-Uk9\\_AI5M32ZYUArh8ha9Dvujpn1WTUf](https://www.youtube.com/playlist?list=PL-Uk9_AI5M32ZYUArh8ha9Dvujpn1WTUf) (дата звернення: 26.04.2025).
12. Ramage K. AutoLISP Quick Start. CADTutor. URL:  
<https://www.cadtutor.net/tutorials/autolisp/quick-start.php> (дата звернення: 26.04.2025).
13. Mac L. Lee Mac Programming. AutoLISP & Visual LISP Tutorials. URL:  
<https://www.lee-mac.com/tutorials.html#dcltutorials> (дата звернення: 28.04.2025).
14. Sanders J. JefferyPSanders. The Ultimate AutoLisp Tutorial. URL:  
<https://www.jefferypsanders.com/autolisptut.html> (дата звернення: 28.04.2025).
15. The Swamp. Visual DCL Programming. URL:  
<https://www.theswamp.org/index.php?board=2.0> (дата звернення: 28.04.2025).

## ДОДАТКИ

### ДОДАТОК А. КОД ПРОГРАМИ

*cabinet.lsp:*

```
(defun c:connectedCabinet (/)

;Встановлення орієнтації
(setq p_ortho (getvar "ORTHOMODE")
      p_snapm (getvar "SNAPMODE")
      p_osmode (getvar "OSMODE")
      p_3dosmode (getvar "3DOSMODE"))

;ORTHO, SNAP, OSNAP
(setvar "ORTHOMODE" 0)
(setvar "SNAPMODE" 0)
(setvar "OSMODE" 0)
(setvar "3DOSMODE" 0)

(vl-cmdf "_UCS" "_World")
(vl-cmdf "_.-view" "_sei")
(vl-cmdf "_vscurrent" "_c")
(vl-cmdf "_grid" "_off")

; --- ActiveX ---
(vl-load-com)
(setq acadApp (vlax-get-acad-object))
(setq doc (vla-get-ActiveDocument acadApp))
```

```

(setq model (vla-get-ModelSpace doc))
; --- AddBox ---
(defun AddBox (pt dx dy dz / px py pz cx cy cz)
  (setq px (car pt) py (cadr pt) pz (caddr pt))
  (setq cx (+ px (/ dx 2.0))
        cy (+ py (/ dy 2.0))
        cz (+ pz (/ dz 2.0)))
  (vla-AddBox model (vlax-3D-point (list cx cy cz)) dx dy dz)
)

; --- Завантаження діалогового вікна---
(if (< (setq dcl_id (load_dialog (findfile "D:\\diploma\\cabinet_form.dcl"))) 0)
  (exit)
)

(if (not (new_dialog "cabinetDialog" dcl_id))
  (exit)
)

; --- Значення за замовчуванням ---
(foreach pair
  '(("width" . "1220") ("depth" . "600") ("height" . "1610") ("thickness" . "18")
    ("h1" . "420")
    ("width_tb" . "1320") ("depth_tb" . "630") ("height_tb" . "710") ("h_verh" . "300")
    ("h_nyz" . "410") ("thickness_tb" . "18") ("pid_vysota" . "70"))
  (set_tile (car pair) (cdr pair))

; --- За замовчуванням для спливаючого вікна кромка---
(set_tile "kromkaChoice" "2"); index 2 = "1 мм"

```

; --- За замовчуванням для прапорця «Зберегти у файл»---

```
(set_tile "saveToFile" "1") ; 1 = true (turned on)
```

```
)
```

```
(setq done nil)
```

:: --- Обробка кнопки «ОК» ---

```
(action_tile
```

```
"accept"
```

```
"(progn
```

```
(setq width (atof (get_tile \"width\")))
```

```
depth (atof (get_tile \"depth\"))
```

```
height (atof (get_tile \"height\"))
```

```
thickness (atof (get_tile \"thickness\"))
```

```
h1 (atof (get_tile \"h1\"))
```

```
width_tb (atof (get_tile \"width_tb\"))
```

```
depth_tb (atof (get_tile \"depth_tb\"))
```

```
height_tb (atof (get_tile \"height_tb\"))
```

```
h_verh (atof (get_tile \"h_verh\"))
```

```
h_nyz (atof (get_tile \"h_nyz\"))
```

```
thickness_tb (atof (get_tile \"thickness_tb\"))
```

```
pid_vysota (atof (get_tile \"pid_vysota\"))
```

```
)
```

```
(setq kromkaIndex (atoi (get_tile \"kromkaChoice\")))
```

```
(setq kromka (nth kromkaIndex '(0.4 0.8 1.0 2.0 3.0)))
```

```
(setq b1 (if (= (get_tile \"saveToFile\") \"1\") 1 0))
```

```
(if (or
```

```
(< width 400) (> width 2400)
```

```
(< depth 300) (> depth 1000)
```

```

(< height 800) (> height 2700)
(< thickness 10) (> thickness 30)
(< h1 100) (> h1 1200)
(< width_tb 400) (> width_tb 2000)
(< depth_tb 300) (> depth_tb 1000)
(< height_tb 300) (> height_tb 1200)
(< h_verh 100) (> h_verh 1000)
(< h_nyz 100) (> h_nyz 1000)
(< pid_vysota 30) (> pid_vysota 500)
(< thickness_tb 10) (> thickness_tb 30)
)

```

```

(alert "\"Перевірте розміри. Ви ввели недопустимі значення або вийшли за межі
допустимного значення.\")

```

```

(progn
  (setq done t)
  (done_dialog 1))
)
)"
)
; --- Обробка кнопки «Скасувати» ---
(action_tile "cancel" "(setq done nil) (done_dialog)")

; --- Запуск DCL ---
(start_dialog)
(unload_dialog dcl_id)
(if (not done)
  (progn (prompt "\nСкасовано. ") (exit))
)
(princ)

```

; === Статичні змінні ===

(setq bodyHeight height) ;

(setq archHeight 280.0)

=====

(setq xyz (getpoint "Будь ласка, введіть передній лівий кут столу "))

(setq x (car xyz)

  y (cadr xyz)

  z (nth 2 xyz))

=====

;Частина 1. Побудова шафи.

;Ліва панель

(setq t1sf (list x y z))

(AddBox t1sf thickness depth bodyHeight)

(setq b\_leftPanel (list bodyHeight depth)) ; ---- вивід

;Права панель

(setq t2sf (list (+ x (- width thickness)) y z))

(AddBox t2sf thickness depth bodyHeight)

(setq b\_rightPanel (list bodyHeight depth)) ; ---- вивід

;Задня панель

(setq t3sf (list (+ x thickness) (+ y (- depth thickness)) z))

(Addbox t3sf (- width (\* 2 thickness)) thickness bodyHeight)

(setq b\_backPanel (list bodyHeight (- width (\* 2 thickness)))) ; --- vuvид

;Нижня панель

```
(setq t4sf (list (+ x thickness) (+ y thickness) z))  
(Addbox t4sf (- width (* 2 thickness)) (- depth (* 2 thickness)) thickness)  
(setq b_bottom (list depth (- width (* 2 thickness)))) ;--- vuvud
```

;Верхня панель

```
(setq t5sf (list (+ x thickness) (+ y thickness) (+ z (- bodyHeight thickness))))  
(Addbox t5sf (- width (* 2 thickness)) (- depth (* 2 thickness)) thickness)  
(setq b_top (list depth (- width (* 2 thickness)))) ;----vuvud
```

;Нижня полицка

```
(setq t6sf (list (+ x thickness) (+ y thickness) (+ z h1)))  
(Addbox t6sf (- width (* 2 thickness)) (- depth (* 2 thickness)) thickness)  
(setq b_shelf1 (list depth (- width (* 2 thickness)))) ;--- vuvud
```

;Вертикальна перегородка під нижньою полицею ( по центру)

```
(setq t7sf (list (+ x (/ width 2.0) (- (/ thickness 2.0))) (+ y thickness) (+ z thickness)))  
(Addbox t7sf thickness (- depth (* 2 thickness)) (- h1 thickness))  
(setq b_vertykalshelf (list (- h1 thickness) (- depth (* 2 thickness)))) ;---- vuvud
```

; Вирівнювання середніх полиць

```
(setq stepHeight (/ (- bodyHeight h1 thickness) 3.0))
```

;Середня нижня полицка

```
(setq t8sf (list (+ x thickness) (+ y thickness) (+ z (+ h1 stepHeight))))  
(Addbox t8sf (- width (* 2 thickness)) (- depth (* 2 thickness)) thickness)  
(setq b_shelf2 (list depth (- width (* 2 thickness)))) ;--- vuvud
```

; Середня верхня полицка

```
(setq t9sf (list (+ x thickness) (+ y thickness) (+ z (+ h1 (* 2 stepHeight)))))
```

```
(Addbox t9sf (- width (* 2 thickness)) (- depth (* 2 thickness)) thickness)
```

```
(setq b_shelf3 (list depth (- width (* 2 thickness)))) ; --- vuvud
```

=====

; Частина 2. Побудова дверей шафи

```
(setq doorWidth (/ (- width (* 2 thickness)) 2.0))
```

```
(setq doorHeight h1)
```

```
(setq fullDoorHeight (- bodyHeight h1 thickness)) ; Висота від верхньої полиці до верху
```

; Ліва нижня дверка

```
(setq t1dr (list (+ x thickness) y (+ z thickness)))
```

```
(Addbox t1dr doorWidth thickness doorHeight)
```

```
(setq b_doorL (list doorHeight doorWidth)) ; ---- vuvud
```

; Права нижня дверка

```
(setq t2dr (list (+ x thickness doorWidth) y (+ z thickness)))
```

```
(Addbox t2dr doorWidth thickness doorHeight)
```

```
(setq b_doorR (list doorHeight doorWidth)) ; --- vuvud
```

; Ліва верхня дверка

```
(setq t3dr (list (+ x thickness) y (+ z h1 thickness)))
```

```
(Addbox t3dr doorWidth thickness fullDoorHeight)
```

```
(setq b_doorTopL (list fullDoorHeight doorWidth)) ; ---- vuvud
```

; Права верхня дверка

```
(setq t4dr (list (+ x thickness fullDoorWidth) y (+ z h1 thickness)))  
(Addbox t4dr doorWidth thickness fullDoorHeight)  
(setq b_doorTopR (list fullDoorHeight doorWidth));---- vuvld
```

=====

```
;Частина 3. Побудова арки
```

```
(vl-cmdf "_UCS" "_Front")
```

```
(setq x1 x  
      y1 (+ z bodyHeight )  
      z1 (* -1 (+ y depth)))
```

```
; --- Зовнішня арка ---
```

```
(setq t1o (list x1 y1 z1)) ;Ліва нижня точка  
(setq t2o (list (+ x1 width) y1 z1)) ;Права нижня точка  
(setq t3o (list x1 (+ y1 0.01) (+ z1 archHeight))) ;Ліва верхня точка  
(setq t4o (list (+ x1 width) (+ y1 0.01) (+ z1 archHeight))) ;Права верхня точка
```

```
(vl-cmdf "_PLINE" t1o t3o "_A" t4o "_L" t2o "_C")
```

```
(setq archPline (entlast))
```

```
(vl-cmdf "_EXTRUDE" archPline "" depth "")
```

```
(setq archSolid (entlast))
```

```
; --- Внутрішня арка ---
```

```
(setq t1i (list (+ x1 thickness) y1 (+ z1 thickness)))  
(setq t2i (list (- (+ x1 width) thickness) y1 (+ z1 thickness)))  
(setq t3i (list (+ x1 thickness) (+ y1 0.01) (+ z1 (- archHeight thickness))))
```

```
(setq t4i (list (- (+ x1 width) thickness) (+ y1 0.01) (+ z1 (- archHeight thickness))))
```

```
(vl-cmdf "_PLINE" t1i t3i "_A" t4i "_L" t2i "_C")
```

```
(setq innerPline (entlast))
```

```
(vl-cmdf "_EXTRUDE" innerPline "" depth "")
```

```
(setq innerArch (entlast))
```

```
(vl-cmdf "_SUBTRACT" archSolid "" innerArch "")
```

```
(setq finalArch archSolid)
```

```
(setq b_arch (list archHeight width)) ; --- vuvud
```

```
(vl-cmdf "_UCS" "_World")
```

=====

```
;Частина 4. Побудова тумбочки
```

```
;Початкова точка тумбочки
```

```
(setq xt (+ x width))
```

```
:: --- Ліва панель ---
```

```
(setq t1tb (list xt y z))
```

```
(AddBox t1tb thickness_tb depth_tb height_tb)
```

```
(setq tb_leftPanel (list height_tb depth_tb)); ----vuvud
```

```
:: --- Права панель ---
```

```
(setq t2tb (list (+ xt (- width_tb thickness_tb)) y z))
```

```
(AddBox t2tb thickness_tb depth_tb height_tb)
```

```
(setq tb_rightPanel (list height_tb depth_tb)); --- vuvud
```

```
:: --- Задня панель ---
```

```
(setq t3tb (list (+ xt thickness_tb) (+ y (- depth_tb thickness_tb)) z))
(AddBox t3tb (- width_tb (* 2 thickness_tb)) thickness_tb height_tb)
(setq tb_backPanel (list height_tb (- width_tb (* 2 thickness_tb)))); --- vuvud
```

```
:: --- Нижня платформа ---
```

```
(setq t4tb (list (+ xt thickness_tb) (+ y thickness_tb) (+ z (- pid_vysota thickness_tb))))
(AddBox t4tb (- width_tb (* 2 thickness_tb)) (- depth_tb (* 2 thickness_tb)) thickness_tb)
(setq tb_bottomPlatform (list depth_tb (- width_tb (* 2 thickness_tb)))); --- vuvud
```

```
:: --- Середня полиця ---
```

```
(setq t5tb (list (+ xt thickness_tb) (+ y thickness_tb) (+ z h_nyz)))
(AddBox t5tb (- width_tb (* 2 thickness_tb)) (- depth_tb (* 2 thickness_tb)) thickness_tb)
(setq tb_middleShelf (list depth_tb (- width_tb (* 2 thickness_tb)))); --- vuvud
```

```
:: --- Верхня панель ---
```

```
(setq t6tb (list (+ xt thickness_tb) (+ y thickness_tb) (+ z (- height_tb thickness_tb))))
(AddBox t6tb (- width_tb (* 2 thickness_tb)) (- depth_tb (* 2 thickness_tb)) thickness_tb)
(setq tb_topPanel (list depth_tb (- width_tb (* 2 thickness_tb)))); --- vuvud
```

```
=====
```

```
;Частина 5. Двері тумбочки
```

```
; --- Передня верхня панель ---
```

```
(setq doorUpperHeight (- height_tb h_nyz thickness_tb)) ; The space between middle shelf  
and top panel
```

```
(setq t7tb (list (+ xt thickness_tb) y (+ z h_nyz thickness_tb))) ; Start after middle shelf  
thickness
```

```
(AddBox t7tb (- width_tb (* 2 thickness_tb)) thickness_tb doorUpperHeight)
```

```
(setq tb_frontTop (list doorUpperHeight (- width_tb (* 2 thickness_tb)))); --- vuvud
```

; --- Передня нижня панель ---

```
(setq doorLowerHeight (+ (- h_nyz pid_vysota) thickness_tb))
```

```
(setq t8tb (list (+ xt thickness_tb) y (+ z (- pid_vysota thickness_tb))))
```

```
(AddBox t8tb (- width_tb (* 2 thickness_tb)) thickness_tb doorLowerHeight)
```

```
(setq tb_frontDoor (list doorLowerHeight (- width_tb (* 2 thickness_tb)))) ;---- vuvid
```

=====

```
(setq b1 1) ; прапорець для активації запису
```

```
(setq vuvid "D:/diploma/cabinet_output.txt")
```

;; -- Розрахунок розмірів деталей з/без кромки ---

; Значення за замовчуванням

```
(defun obrizka1 (len / ob_x ob_y)
```

```
(setq ob_x (abs (car len))
```

```
      ob_y (abs (cadr len)))
```

```
(list ob_x ob_y)
```

```
)
```

;Кромка попереду

```
(defun obrizka_kromku1 (len / )
```

```
(setq ob_x (- (abs (car len)) kromka)
```

```
      ob_y (abs (cadr len)))
```

```
(setq vuvid_kr1 (list ob_x ob_y))
```

```
)
```

;Кромка зверху

```
(defun obrizka_kromku2 (len / )
```

```
(setq ob_x (abs (car len))
```

```
      ob_y (- (abs (cadr len)) kromka))
```

```
(setq vuvid_kr2 (list ob_x ob_y))  
)
```

;Кромка попереду і зверху

```
(defun obrizka_kromku12 (len / )  
(setq ob_x (- (abs (car len)) kromka)  
      ob_y (- (abs (cadr len)) kromka))  
(setq vuvid_kr12 (list ob_x ob_y))  
)
```

;Кромка для всіх сторін – для дверей

```
(defun obrizka_kromkuAll (len / ob_x ob_y)  
(setq ob_x (- (abs (car len)) (* 2 kromka))  
      ob_y (- (abs (cadr len)) (* 2 kromka)))  
(list ob_x ob_y)  
)
```

:: --- Запис у файл ---

```
(if (= b1 1)
```

```
  (progn
```

```
    (setq desk (open (strcat vuvid) "w"))
```

:: --- З урахуванням личківки ---

```
(write-line "Розміри заготовок шафки та тумбочки:" desk)
```

```
(write-line "\nРозміри деталей шафи" desk)
```

```
(write-line (strcat "1. Ліва панель шафи: " (vl-prin1-to-string b_leftPanel)) desk)
```

```
(write-line (strcat "2. Права панель шафи: " (vl-prin1-to-string b_rightPanel)) desk)
```

```
(write-line (strcat "3. Задня панель шафи: " (vl-prin1-to-string b_backPanel)) desk)
```

```
(write-line (strcat "4. Верхня панель шафи: " (vl-prin1-to-string b_top)) desk)
```

```
(write-line (strcat "5. Нижня панель шафи: " (vl-prin1-to-string b_bottom)) desk)
(write-line (strcat "6. Нижня поличка: " (vl-prin1-to-string b_shelf1)) desk)
(write-line (strcat "7. Вертикальна поличка-перегородка: " (vl-prin1-to-string
b_vertikalshelf)) desk)
(write-line (strcat "8. Середня нижня поличка: " (vl-prin1-to-string b_shelf2)) desk)
(write-line (strcat "9. Середня верхня поличка: " (vl-prin1-to-string b_shelf3)) desk)
(write-line (strcat "10. Нижні двері ліві: " (vl-prin1-to-string b_doorL)) desk)
(write-line (strcat "11. Нижні двері праві: " (vl-prin1-to-string b_doorR)) desk)
(write-line (strcat "12. Верхні двері ліві: " (vl-prin1-to-string b_doorTopL)) desk)
(write-line (strcat "13. Верхні двері праві: " (vl-prin1-to-string b_doorTopR)) desk)
(write-line (strcat "14. Арка шафи: " (vl-prin1-to-string b_arch)) desk)
```

```
(write-line "\nРозміри деталей тумбочки:" desk)
(write-line (strcat "15. Ліва панель тумбочки: " (vl-prin1-to-string tb_leftPanel)) desk)
(write-line (strcat "16. Права панель тумбочки: " (vl-prin1-to-string tb_rightPanel))
desk)
(write-line (strcat "17. Задня панель тумбочки: " (vl-prin1-to-string tb_backPanel))
desk)
(write-line (strcat "18. Нижня платформа тумбочки: " (vl-prin1-to-string
tb_bottomPlatform)) desk)
(write-line (strcat "18. Середня поличка тумбочки: " (vl-prin1-to-string
tb_middleShelf)) desk)
(write-line (strcat "20. Верхня панель тумбочки: " (vl-prin1-to-string tb_topPanel))
desk)
(write-line (strcat "21. Верхня двері тумбочки: " (vl-prin1-to-string tb_frontTop)) desk)
(write-line (strcat "22. Нижні двері тумбочки: " (vl-prin1-to-string tb_frontDoor)) desk)
```

```
:: --- БЕЗ урахування личківки ---
```

```
(write-line "\nРозміри деталей шафи БЕЗ урахування личківки:" desk)
(write-line (strcat "1. Ліва панель шафи: " (vl-prin1-to-string (obrizka_kromku1
b_leftPanel)))) desk)
(write-line (strcat "2. Права панель шафи: " (vl-prin1-to-string (obrizka_kromku1
b_rightPanel)))) desk)
(write-line (strcat "3. Задня панель шафи: " (vl-prin1-to-string (obrizka1 b_backPanel)))
desk)
(write-line (strcat "4. Верхня панель шафи: " (vl-prin1-to-string (obrizka_kromku1
b_top)))) desk)
(write-line (strcat "5. Нижня панель шафи: " (vl-prin1-to-string (obrizka_kromku1
b_bottom)))) desk)
(write-line (strcat "6. Нижня поличка: " (vl-prin1-to-string (obrizka_kromku1
b_shelf1)))) desk)
(write-line (strcat "7. Вертикальна поличка-перегородка: " (vl-prin1-to-string
(obrizka_kromku1 b_vertykalshelf)))) desk)
(write-line (strcat "8. Середня нижня поличка: " (vl-prin1-to-string (obrizka_kromku1
b_shelf2)))) desk)
(write-line (strcat "9. Середня верхня поличка: " (vl-prin1-to-string (obrizka_kromku1
b_shelf3)))) desk)
(write-line (strcat "10. Нижні двері ліві: " (vl-prin1-to-string (obrizka_kromkuAll
b_doorL)))) desk)
(write-line (strcat "11. Нижні двері праві: " (vl-prin1-to-string (obrizka_kromkuAll
b_doorR)))) desk)
(write-line (strcat "12. Верхні двері ліві: " (vl-prin1-to-string (obrizka_kromkuAll
b_doorTopL)))) desk)
(write-line (strcat "13. Верхні двері праві: " (vl-prin1-to-string (obrizka_kromkuAll
b_doorTopR)))) desk)
(write-line (strcat "14. Арка шафи: " (vl-prin1-to-string (obrizka_kromku1 b_arch)))
desk)
(write-line "\nРозміри деталей тумбочки БЕЗ урахування личківки:" desk)
```

```

(write-line (strcat "15. Ліва панель тумбочки: " (vl-prin1-to-string (obrizka_kromku12
tb_leftPanel)))) desk)
(write-line (strcat "16. Права панель тумбочки: " (vl-prin1-to-string (obrizka_kromku12
tb_rightPanel)))) desk)
(write-line (strcat "17. Задня панель тумбочки: " (vl-prin1-to-string (obrizka_kromku2
tb_backPanel)))) desk)
(write-line (strcat "18. Нижня панель тумбочки: " (vl-prin1-to-string (obrizka_kromku1
tb_bottomPlatform)))) desk)
(write-line (strcat "19. Середня поличка тумбочки: " (vl-prin1-to-string
(obrizka_kromku1 tb_middleShelf)))) desk)
(write-line (strcat "20. Верхня панель тумбочки: " (vl-prin1-to-string (obrizka_kromku1
tb_topPanel)))) desk)
(write-line (strcat "21. Верхні передні двері тумбочки: " (vl-prin1-to-string
(obrizka_kromkuAll tb_frontTop)))) desk)
(write-line (strcat "22. Нижні передні двері тумбочки: " (vl-prin1-to-string
(obrizka_kromkuAll tb_frontDoor)))) desk)
(close desk)
(prompt (strcat "\nДані записано у файл: " vuvid))
)
)
; Відновлення стану режимів ORTHO, SNAP , OSNAP
(setvar "ORTHOMODE" p_ortho)
(setvar "SNAPMODE" p_snapm)
(setvar "OSMODE" p_osmode)
(setvar "3DOSMODE" p_3dosmode)

(princ)
)

```

***cabinet\_form.dcl:***

```

cabinetDialog : dialog {
  label = "Додаток AutoCAD для побудови шафи та тумбочки";
  : column {
    : boxed_column {
      label = "Шафа";
      : row { : text { label = "Ширина:";      alignment = left; } : edit_box { key = "width";
edit_width = 30; } }
      : row { : text { label = "Глибина:";      alignment = left; } : edit_box { key = "depth";
edit_width = 30; } }
      : row { : text { label = "Висота:";      alignment = left; } : edit_box { key = "height";
edit_width = 30; } }
      : row { : text { label = "Нижня полиця:";      alignment = left; } : edit_box { key =
"h1";      edit_width = 30; } }
      : row { : text { label = "Товщина:";      alignment = left; } : edit_box { key =
"thickness"; edit_width = 30; } }
    }
    : boxed_column {
      label = "Тумбочка";
      : row { : text { label = "Ширина:";      alignment = left; } : edit_box { key =
"width_tb";  edit_width = 30; } }
      : row { : text { label = "Глибина:";      alignment = left; } : edit_box { key =
"depth_tb";  edit_width = 30; } }
      : row { : text { label = "Висота:";      alignment = left; } : edit_box { key =
"height_tb"; edit_width = 30; } }
      : row { : text { label = "Верхня полиця:"; alignment = left; } : edit_box { key =
"h_verh";    edit_width = 30; } }
      : row { : text { label = "Середня полиця:"; alignment = left; } : edit_box { key = "h_nyz";
edit_width = 30; } }
      : row { : text { label = "Нижня полиця:";      alignment = left; } : edit_box { key =
"pid_vysota"; edit_width = 30; } }
    }
  }
}

```

```

    : row { : text { label = "Товщина:";      alignment = left; } : edit_box { key =
"thickness_tb"; edit_width = 30; } }
}
: boxed_column {
label = "Налаштування";
: row {
: text { label = "Товщина кромки:"; alignment = left; }
: popup_list {
key = "kromkaChoice";
edit_width = 20;
value = "1 мм";
list = "0,4 мм\n0,8 мм\n1 мм\n2 мм\n3 мм";
}
}
: row {
: toggle {
label = "Записати дані у файл";
key = "saveToFile";
value = "0";
}
}
}
: row {
: button { label = "ОК"; key = "accept"; is_default = true; }
: button { label = "Скасувати"; key = "cancel"; is_cancel = true; }
}
}
}

```