

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та  
комп'ютерних технологій і дизайну  
(повне найменування інституту, назва факультету (відділення))

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

## Пояснювальна записка

до дипломної роботи

ОКР – бакалавр

(освітньо-кваліфікаційний рівень)

на тему: Розроблення САПР меблів для передпокою засобами AutoCAD.

Виконав: студент 2 курсу групи ICTC-21  
спеціальності

126 “Інформаційні системи та технології”

(шифр і назва напрямку підготовки, спеціальності)

Цвик А.Л.

(прізвище та ініціали)

Керівник Сторожук О.Л.

(прізвище та ініціали)

Рецензент Крошніий І.М.

(прізвище та ініціали)

Львів – 2023

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

ННІ деревообробних та комп'ютерних технологій і дизайну  
Кафедра інформаційних систем та комп'ютерного моделювання  
Рівень вищої освіти перший (бакалаврський)  
Спеціальність 126 "Інформаційні системи та технології"  
(шифр і назва)

**ЗАТВЕРДЖУЮ**

В.о.завідувача кафедри

Сторожук О.Л.

" 12 " 06 2023 року

**ЗАВДАННЯ**  
**НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Цвик Анастасії Любомирівні

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення САПР меблів для передпокою засобами AutoCAD»  
керівник роботи Сторожук Олександр Леонідович, к.т.н., доцент

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

затверджені наказом вищого навчального закладу від "21" 11 2022 року № С-521

2. Термін подання студентом роботи 12.06.2023р.

3. Вихідні дані до роботи: Постановка задачі та її формалізація. Основні параметри та вимоги до проектування корпусних меблів.

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 креслень)

Презентація до диплому

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 23 листопада 2022 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд літературних даних.	23.11.2022р. 21.12.2022р.	– викон.
2.	Розділ 1. Стан проблемної області.	21.12.2022р. 05.01.2023р.	– викон.
3.	Розділ 2. Інформаційне та математичне забезпечення.	05.01.2023р. 02.02.2023р.	– викон.
4.	Розділ 3. Програмне та технічне забезпечення.	02.02.2023р. 12.05.2023р.	– викон.
9.	Аналіз отриманих результатів та написання висновків. Оформлення дипломної роботи.	12.05.2023р. 06.06.2023р.	– викон.
10.	Здача пояснювальної записки на перевірку керівнику, виправлення помилок та здача роботи рецензенту.	12.06.2023 р.	викон.

Студент

Керівник роботи

  
(підпис)

  
(підпис)

Цвик А.Л.

(прізвище та ініціали)

Сторожук О.Л.

(прізвище та ініціали)

## РЕФЕРАТ

Дипломна робота містить 53 сторінок пояснювальної записки, 5 рисунків, 1 додаток, 15 джерел.

Проектування меблів для передпокою реалізовано на прикладі передпокою “Грейд Комфорт”, що представлений у торговій мережі Епіцентр.

Під час проектування та створення програмного додатку використовувались мови програмування AutoLISP, VisualLisp та DCL. Результатом роботи розробленої САПР є геометрична 3D побудова передпокою “Грейд Комфорт”. Реалізовано виведення усіх результатів геометричних розрахунків у файл для подальшого створення необхідної документації.

*Ключові слова:* об’єкт, функція, AutoLisp, VisualLisp, DCL, алгоритм, програмний код.

## ABSTRACT

The thesis contains 53 pages of an explanatory note, 5 figures, 1 appendix, 15 sources.

The design of furniture for the hallway is implemented using the example of the Grade Comfort hallway, presented in the Epicenter shopping network.

During the design and creation of the software application, programming languages AutoLISP, VisualLisp and DCL were used. The result of the work of the developed CAD is the geometric 3D construction of the Grade Comfort hallway. All results of geometric calculations are displayed in the file for further creation of the necessary documentation.

*Keywords:* object, function, AutoLisp, VisualLisp, DCL, algorithm, program code.

## ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити програмне забезпечення проектування меблів для передпокою. Програмне забезпечення має бути розроблено у вигляді додатка для AutoCAD2023, написане мовами програмування AutoLISP, VisualLisp та DCL в середовищі:

- VisualLISP for AutoCAD;  
з наглядним та зрозумілим графічним інтерфейсом.

Графічний інтерфейс реалізувати за допомогою мови DCL. Передбачити виведення усіх результатів геометричних розрахунків у файл для подальшого створення необхідної документації.

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

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

САПР – система автоматизованого проектування;

ІЗ – інформаційне забезпечення;

LISP – LIST Processor (обробник списків);

AutoLISP – діалект мови програмування LISP, в який добавлено функції доступу до об'єктів, таблиць та словників системи AutoCAD;

ОС – операційна система;

ПЗ – програмне забезпечення.

**ЗМІСТ**

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ .....	6
ЗМІСТ .....	7
ВСТУП.....	8
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ .....	9
1.1. Огляд проблемної області. ....	9
1.2. Які системи САПР існують для проектування меблів .....	11
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	22
2.1. Мови LISP, AutoLISP та VisualLISP.....	22
2.2. Засоби інформаційного забезпечення .....	32
2.3. Засоби математичного забезпечення.....	39
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	47
3.1. Основні відомості про AutoCAD .....	47
3.2. Реалізація програмного забезпечення .....	48
ВИСНОВКИ .....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ .....	57

## ВСТУП

Сьогодні віртуальна реальність та інші інноваційні технології займають все більш вагому роль у житті людей, зокрема в дизайні та розробці меблів. Від сучасних програм для проектування меблів відомий широкий вибір функцій, які дозволяють розробникам швидко та ефективно проектувати нові моделі та оптимізувати вже існуючі.

Ціль даного дипломного проекту полягає у розробці системи автоматизованого проектування меблів для передпокою з використанням програми AutoCAD. Програма AutoCAD є однією з найбільш популярних у своєму роді та має великий потенціал для розробки САПР. Розроблена система дозволяє створювати та модифікувати меблі для передпокою за допомогою графічного інтерфейсу, що значно спростить роботу дизайнерів та розробників меблів.

У даній роботі буде описано процес розробки САПР меблів для передпокою, зокрема від проектування системи до розробки графічного інтерфейсу користувача. Крім того, буде проведено дослідження функціональності та ефективності розробленої системи та проведений порівняльний аналіз із іншими програмами для проектування меблів.

Отже, дана робота є актуальною та важливою з точки зору розвитку інноваційних технологій та дизайну меблів, що може знайти застосування у багатьох галузях.

**Об'єкт дослідження** – засоби створення програмного додатку до AutoCAD 2023.

**Предмет дослідження** – мови програмування AutoLISP, VisualLisp та DCL.

**Мета роботи** – розроблення САПР меблів для передпокою у вигляді програмного додатку до AutoCAD 2023.

**Практичне значення** – розроблений додаток значно пришвидшує проектування, виготовлення та візуалізацію виробу.

## РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1. Огляд проблемної області.

Меблева галузь в Україні є однією з найбільш проблемних сфер економіки. Незважаючи на те, що в Україні виробляється велика кількість меблів, існує низка факторів, які перешкоджають розвитку галузі і знижують її конкурентоспроможність на міжнародному ринку.

Однією з основних проблем є відсутність якісних матеріалів та комплектуючих, які б відповідали світовим стандартам. Часто виробники меблів змушені використовувати дешеві матеріали, що негативно впливає на якість продукції та безпеку користування меблями.

Також важко знайти кваліфіковану робочу силу для виробництва меблів. Багато молодих людей віддають перевагу іншим сферам діяльності, що призводить до дефіциту робочої сили в галузі.

Іншою проблемою є низький рівень інноваційності виробників меблів. Вони часто не використовують нові технології та не вивчають нові ринки, що призводить до втрати конкурентоспроможності. Крім того, в Україні існує велика кількість нелегальних виробників меблів, які виготовляють продукцію низької якості та порушують законодавство у сфері виробництва меблів.

У зв'язку з цим, влада та бізнес повинні спільно працювати над вирішенням цих проблем, зокрема, сприяти розвитку виробництва якісних матеріалів та комплектуючих, підвищенню кваліфікації робочої сили, стимулюванню інноваційності в галузі та боротьбі з нелегальним виробництвом меблів.

Окрім цього, важливим кроком до розвитку меблевої галузі є підвищення якості продукції та її дизайну, щоб привернути увагу покупців на внутрішньому та зовнішньому ринках. Для цього необхідно віддавати перевагу сучасним технологіям виробництва та використовувати інноваційні матеріали.

Також варто активно працювати над підвищенням ефективності меблевого виробництва та його логістики, що дозволить знизити витрати на виробництво та

збільшити його рентабельність. Для цього можна використовувати сучасні технології автоматизації та впроваджувати інноваційні підходи до управління бізнесом.

Додатковим кроком до поліпшення стану меблевої галузі в Україні може бути посилення сприяння держави шляхом введення спеціальних програм та пільгових умов для меблевих підприємств. Це може включати зниження податкового тягаря, надання фінансової підтримки, залучення інвестицій, сприяння відкриттю нових ринків збуту та експорту.

Крім того, важливо розвивати меблевий дизайн як окрему галузь. Підтримка інноваційних дизайнерських рішень та впровадження сучасних тенденцій в меблевому виробництві можуть допомогти привернути увагу до українських меблів на світовому ринку. Залучення професійних дизайнерів і співпраця з ними можуть значно підвищити якість та естетичний вигляд українських меблів [1].

Також варто відзначити значний потенціал розвитку e-commerce у меблевій галузі. Інтернет-торгівля меблями набуває все більшої популярності, тому важливо підтримувати розвиток онлайн-продажів та створювати зручні та безпечні платформи для покупців.

Загалом, розвиток меблевої галузі в Україні потребує комплексного підходу та взаємодії різних сторін, що дозволить збільшити її конкурентоспроможність та підвищити якість виробленої продукції. Тільки таким чином меблева галузь зможе стати вагомим гравцем на міжнародному ринку та забезпечити стабільний розвиток економіки країни.

Україна також може використовувати свої природні ресурси, зокрема деревину, як конкурентну перевагу в меблевому виробництві. Розвиток екологічно чистих технологій та виробництва деревинних меблів може сприяти підвищенню попиту на українську продукцію [12].

Окрім внутрішніх факторів, меблева галузь України також стикається з рядом зовнішніх викликів і проблем. Однією з них є конкуренція з боку імпортованих меблів. Високі митні тарифи, бюрократичні процедури та низька

конкурентоспроможність українських виробників у порівнянні з зарубіжними компаніями становлять значний виклик для розвитку меблевої галузі.

Крім того, нестабільна політична та економічна ситуація в країні, а також недостатня захищеність прав власності та корупція можуть також ускладнювати діяльність меблевих підприємств. Ці фактори можуть гальмувати інвестиції та стимулювати нелегальну діяльність у галузі.

Також варто зазначити, що велика частина меблевого ринку в Україні зосереджена на середньому та нижчому сегменті, виробляючи меблі масового виробництва. Недостатня увага до високоякісних та дизайнерських меблів обмежує можливості галузі у просуванні на більш преміальний ринок.

Загалом, для подолання цих проблем необхідна комплексна стратегія, яка включатиме в себе підтримку від держави, створення сприятливих умов для бізнесу, стимулювання інновацій, просування української меблевої продукції на зовнішні ринки, підвищення якості та дизайну, а також залучення іноземних інвестицій. Тільки шляхом спільних зусиль всіх зацікавлених сторін можна досягти покращення стану меблевої галузі в Україні [2].

## **1.2. Які системи САПР існують для проектування меблів**

Системи САПР є потужними інструментами, що дозволяють дизайнерам, інженерам та виробникам меблів створювати, моделювати та аналізувати різні типи меблів у віртуальному середовищі [11]. Вони сприяють покращенню ефективності, точності та швидкості проектування, а також дозволяють вирішувати складні завдання, пов'язані з конструкцією та виробництвом меблів.

Існує широкий спектр систем САПР, що включають AutoCAD, SketchUp, SolidWorks, 3ds Max, Fusion 360, Chief Architect, Solid Edge, Cabinet Vision, Promob, SolidWorks Composer, Rhino 3D, Pro100, SolidCAM та інші. Кожна з цих систем має свої унікальні можливості та функціонал, але вони спрямовані на полегшення та ускладнення процесу проектування меблів, в залежності від потреб користувача.

### - **AutoCAD:**

AutoCAD - одна з найпоширеніших систем автоматизованого проектування (САПР), яка знаходить широке застосування в різних галузях, включаючи проектування меблів. Вперше випущена в 1982 році компанією Autodesk, AutoCAD став стандартом у галузі комп'ютерного проектування та малювання.

Основною перевагою AutoCAD є його універсальність та гнучкість. Система надає широкий набір інструментів для створення 2D- та 3D-моделей меблів. Завдяки простому інтерфейсу та потужним функціям, AutoCAD дозволяє проектувати меблі з точністю до деталі, враховуючи різні параметри та вимоги. AutoCAD дозволяє проектувати меблі з точністю до кожної деталі. Вона має широкий набір команд та функцій для створення різних типів меблів, включаючи столи, шафи, ліжка, крісла та багато іншого. Завдяки могутностям AutoCAD, можна створювати складні меблі з різними формами, кутами та деталізацією [13].

AutoCAD має можливості для створення креслень, планів, розробки просторових моделей меблів, додавання текстур та матеріалів, анімації, візуалізації та інших функцій, необхідних для повноцінного проектування меблів. Він також підтримує імпорт та експорт різних форматів файлів, що сприяє обміну даними з іншими системами САПР та співпраці з іншими фахівцями.

AutoCAD надає можливості автоматизації та налаштування, що дозволяє користувачам створювати власні бібліотеки, блоки та шаблони меблів для використання в подальших проектах. Це спрощує роботу та підвищує продуктивність при проектуванні меблів.

### - **SketchUp:**

За допомогою SketchUp виконуються різноманітні задачі, пов'язані з проектуванням і візуалізацією меблів. Основні задачі, які можна вирішити за допомогою SketchUp, включають:

Моделювання меблів: SketchUp дозволяє створювати 3D-моделі меблів з точністю до деталей. Ви можете створювати різноманітні форми меблів, додавати текстури, змінювати розміри та конфігурацію меблів.

Проектування інтер'єру: SketchUp дозволяє створювати 3D-моделі приміщень та виконувати їхнє оздоблення. Ви можете розташовувати моделі меблів у просторі, визначати оптимальні розміри та розміщення меблів у кімнаті.

Візуалізація та рендеринг: SketchUp має інструменти для створення фотореалістичних зображень меблів та інтер'єру. Ви можете застосовувати освітлення, матеріали та текстури для створення реалістичних візуалізацій вашого проекту.

Редагування та модифікація моделей: SketchUp надає можливість легко редагувати та модифікувати створені моделі. Ви можете змінювати розміри, форму, додавати деталі та виконувати різноманітні маніпуляції з об'єктами меблів.

Поділка проекту та співпраця: SketchUp дозволяє ефективно співпрацювати з іншими учасниками проекту. Ви можете обмінюватися моделями, коментарями та ідеями, робити спільну роботу над проектом меблів.

Створення технічної документації: SketchUp надає можливість генерувати креслення, плани та іншу технічну документацію для вашого проекту меблів. Ви можете створювати зрізи, розміркові лінії та інші елементи, необхідні для створення повноцінного набору документації.

Загалом, SketchUp є потужним інструментом для проектування меблів, який надає широкий спектр функцій для моделювання, візуалізації та співпраці. Він дозволяє втілити ваші ідеї у віртуальний простір та зробити їх доступними для спільної роботи та вирішення проектних завдань [3].

#### - **SolidWorks:**

SolidWorks є потужною системою САПР, яка широко використовується в проектуванні меблів та інших виробів. Ця програма спеціалізується на 3D-моделюванні, має розширений набір інструментів та функцій, які дозволяють проектувати складні меблі з точністю до дрібниць.

SolidWorks дозволяє створювати детальні 3D-моделі меблів з використанням різних функцій, таких як витягування, випуклі та вогнуті форми, з'єднання

компонентів та багато інших. Користувачі можуть моделювати окремі деталі меблів, а потім збирати їх у повністю функціональні 3D-моделі.

Однією з особливостей SolidWorks є можливість використовувати різні методи розробки, такі як параметричне моделювання та варіантні конфігурації. Це дозволяє легко змінювати розміри, форми та матеріали меблів, а також швидко створювати різні варіанти дизайну. Крім того, SolidWorks надає можливість розрахунку міцності та стійкості конструкції меблів, що допомагає враховувати технічні аспекти при проектуванні.

SolidWorks також має вбудовані інструменти для створення технічної документації, яка включає креслення, специфікації та розгортки деталей. Це дозволяє забезпечити точні та докладні вказівки для виробництва меблів [4].

#### - **3ds Max:**

3ds Max є відомою системою САПР, яка знайшла широке застосування у проектуванні та візуалізації меблів. Ця програма відкриває безліч можливостей для створення реалістичних 3D-моделей меблів з високою деталізацією та великим рівнем візуального реалізму.

3ds Max забезпечує користувачів широким спектром інструментів та функцій для моделювання меблів. Вона дозволяє створювати складні форми та деталі, використовуючи різні методи, такі як полігональне моделювання, моделювання з використанням поверхонь або моделювання з використанням NURBS. Крім того, 3ds Max надає можливість використовувати текстури, матеріали та освітлення для досягнення реалістичного вигляду меблів.

Однією з ключових переваг 3ds Max є його потужність у візуалізації. Вона дозволяє створювати фотореалістичні зображення та анімації меблів, що допомагає відтворити їх зовнішній вигляд, матеріали та освітлення з високою точністю. Це особливо корисно для презентацій клієнтам або використання у маркетингових матеріалах.

3ds Max також має можливості для імпорту та експорту різних форматів файлів, що дозволяє обмінюватись даними з іншими системами САПР та

співпрацювати з іншими професіоналами у галузі меблевого проектування. Крім того, програма підтримує різні плагіни та розширення, які додають додаткові функціональні можливості та сприяють розширенню можливостей програми.

Загалом, 3ds Max є потужним інструментом для проектування та візуалізації меблів. Вона дозволяє створювати деталізовані 3D-моделі з реалістичним виглядом та візуалізувати їх у високій якості. Завдяки своїм функціям та можливостям, 3ds Max є важливим інструментом для професіоналів у галузі меблевого дизайну та проектування.

- **Fusion 360:**

Fusion 360 є інноваційною системою САПР, яка знайшла широке застосування в проектуванні меблів та інших виробів. Вона поєднує в собі функції 3D-моделювання, інженерного проектування та співпраці в одному інтегрованому середовищі.

Однією з ключових переваг Fusion 360 є його хмарна платформа, яка дозволяє користувачам зберігати, керувати та спільно працювати над своїми проектами з будь-якого пристрою. Це дозволяє командам дизайнерів та інженерів працювати разом над проектами меблів, обмінюватись даними та спільно вирішувати завдання, забезпечуючи ефективну комунікацію та співпрацю.

Fusion 360 надає широкий спектр інструментів для моделювання меблів. Вона дозволяє створювати складні форми, використовуючи технології прямого та параметричного моделювання. Користувачі можуть швидко створювати 3D-моделі меблів, редагувати їх параметри та змінювати їх дизайн за допомогою простого та інтуїтивного інтерфейсу.

Однією з особливостей Fusion 360 є його можливості для інтегрованого проектування та інженерної розробки. Крім моделювання, програма надає інструменти для створення технічної документації, проведення аналізу міцності та стійкості конструкції, а також для генерації CNC-коду для виготовлення меблів.

Fusion 360 також має можливість використовувати плагіни та розширення для розширення функціональності програми. Користувачі можуть використовувати

додаткові інструменти та ресурси, розроблені спільнотою користувачів, що дозволяє налаштувати програму під свої потреби та вимоги.

- **Chief Architect:**

Chief Architect є відомою системою САПР, спеціалізованою на проектуванні і візуалізації меблів та архітектурних споруд. Ця програма надає широкий набір інструментів та функцій, які дозволяють створювати деталізовані 3D-моделі меблів та віртуальні прогулянки по інтер'єру.

Chief Architect дозволяє проектувати меблі шляхом створення точних 2D- та 3D-моделей, використовуючи різноманітні інструменти. Користувачі можуть створювати креслення меблів, розташовувати їх у просторі, змінювати розміри та форму, додавати матеріали та текстури. Багатофункціональні бібліотеки предметів дозволяють вибрати потрібні компоненти меблів з великого асортименту.

Однією з особливостей Chief Architect є можливість візуалізувати проєктовані меблі та інтер'єр за допомогою фотореалістичних зображень та відеопрезентацій. Завдяки потужним інструментам рендерингу та освітлення, користувачі можуть створювати реалістичні візуалізації, які допомагають уявити, як будуть виглядати меблі у реальному просторі.

Chief Architect також підтримує інтеграцію з іншими системами, наприклад, з програмами для створення технічної документації, що дозволяє швидко та зручно створювати плани, креслення та специфікації для виробництва меблів.

- **Solid Edge:**

Solid Edge - це потужна система САПР, розроблена компанією Siemens PLM Software. Вона спеціалізується на проектуванні меблів та виробництві, і має ряд особливостей, які роблять її привабливою для дизайнерів та інженерів:

1. Синхронне моделювання: Solid Edge використовує синхронну технологію моделювання, що дозволяє швидко та ефективно створювати та редагувати 3D-моделі меблів. Ця технологія дозволяє працювати з готовими моделями незалежно від їх походження, що забезпечує гнучкість та ефективність в проектуванні.

2. Інтуїтивний інтерфейс: Solid Edge має зручний та легкий у використанні інтерфейс, що дозволяє користувачам швидко освоїти програму. Він має інтуїтивні панелі інструментів, доступ до команд у кількох клацаннях та можливість налаштування робочого простору під власні потреби.

4. Широкий набір інструментів: Solid Edge пропонує багато інструментів та функцій, які сприяють ефективному проектуванню меблів. Від моделювання деталей та збірки до аналізу міцності та валідації, Solid Edge забезпечує повний набір інструментів для різних етапів проектування.

5. Інтеграція зі зовнішніми системами: Solid Edge має вбудовану підтримку для інтеграції з системами управління даними (PDM) та іншими зовнішніми програмами. Це дозволяє легко обмінюватися даними та співпрацювати з іншими учасниками проекту.

6. Візуалізація та анімація: Solid Edge дозволяє створювати реалістичні візуалізації меблів, а також створювати анімації для відображення рухомих елементів або функціональності меблів.

7. Керування життєвим циклом продукту (PLM): Solid Edge підтримує PLM-функціонал, що дозволяє керувати всім життєвим циклом продукту, включаючи управління версіями, контроль доступу та спільну роботу [4].

#### **- Cabinet Vision:**

Cabinet Vision - це спеціалізована система САПР, призначена для проектування та виробництва меблів, зокрема кухонь. Вона надає рішення для швидкого та ефективного створення моделей меблів, враховуючи індивідуальні вимоги та налаштування.

Однією з ключових особливостей Cabinet Vision є наявність готових бібліотек елементів, що дозволяють швидко складати компоненти меблів з належними параметрами. Програма також надає можливість візуалізувати створені моделі в реалістичному вигляді, що дозволяє оцінити їх зовнішній вигляд та внести необхідні зміни.

Крім того, Cabinet Vision допомагає економити матеріали шляхом оптимізації розташування компонентів на панелях, зменшуючи відходи матеріалу. Вона також забезпечує інтеграцію з CNC-машинами, що дозволяє автоматично передавати дані проекту для обробки матеріалів.

Усі ці особливості роблять Cabinet Vision потужним інструментом для проектування та виробництва меблів, що спрощує процес створення, оптимізує використання матеріалів і покращує якість кінцевих виробів.

- **Promob:**

Promob - це програмне забезпечення, призначене для професіоналів в галузі дизайну меблів та інтер'єрів. Його інтуїтивний і простий у використанні інтерфейс дозволяє легко та швидко створювати дизайнерські проекти. Програма має широкий вибір готових моделей меблів та матеріалів, що дозволяє вам створювати вражаючі візуалізації своїх ідей.

Promob надає можливість робити детальне 3D-моделювання меблів, дозволяючи змінювати їх параметри, матеріали та стилі, щоб отримати бажаний вигляд. Це дозволяє вам створювати віртуальні приміщення з реалістичними меблями, освітленням та декоративними елементами.

Один з головних аспектів Promob - це можливість оптимізувати використання простору. Ви можете розташовувати меблі в приміщеннях таким чином, щоб максимізувати їх функціональність та ергономіку. Програма також надає інструменти для генерації технічних креслень, зрізів та розгорток, що сприяє виробництву меблів згідно з проектом.

Promob є незамінним інструментом для професіоналів, що працюють у галузі дизайну меблів та інтер'єрів. Він допомагає втілити творчі ідеї в реалістичні проекти, оптимізувати використання простору та спрощує комунікацію з клієнтами.

- **SolidWorks Composer:**

SolidWorks Composer - це програмне забезпечення, призначене для створення ілюстрацій та анімацій на основі 3D-моделей з метою забезпечення зрозумілого та

ефективного спілкування технічних даних. Воно дозволяє перетворити складну технічну інформацію в інтерактивні та легко зрозумілі візуальні матеріали.

За допомогою SolidWorks Composer можна створювати візуальні ілюстрації, анімації та відео, що допомагають в поясненні процесів збирання, обслуговування та ремонту. Програма має простий та зручний інтерфейс, що дозволяє легко маніпулювати 3D-моделями, додавати анімацію, текст та інші елементи для більш зрозумілого представлення інформації.

SolidWorks Composer також дозволяє створювати інтерактивні документи, які дають можливість користувачам взаємодіяти з моделями, обертати, масштабувати та детально досліджувати їх, отримуючи додаткову інформацію. Це дозволяє зручно комунікувати та навчатися.

Узагалі, SolidWorks Composer спрощує процес створення технічної документації, забезпечуючи зрозумілі та інтерактивні ілюстрації та анімації для кращого розуміння технічних процесів та продуктів. Він є цінним інструментом для команд, що займаються розробкою, виробництвом та обслуговуванням складних технічних продуктів [4].

#### - **Rhino 3D:**

Rhino 3D - це програмне забезпечення для тривимірного моделювання і дизайну. Воно відоме своєю гнучкістю і потужністю, що дозволяє створювати складні геометричні форми та поверхні [14].

Rhino 3D має інтуїтивний і зручний інтерфейс, що полегшує процес створення 3D-моделей. Він надає різноманітні інструменти для створення, редагування і аналізу геометрії, включаючи криві, поверхні, сітки і тверді тіла.

Одна з основних переваг Rhino 3D - його здатність працювати з різними форматами файлів, що дозволяє взаємодіяти з іншими CAD-системами та програмами дизайну. Ви можете імпортувати та експортувати моделі в таких форматах, як DWG, STL, IGES, STEP та інші.

Rhino 3D також підтримує велику кількість додаткових плагінів і розширень, які розширюють можливості програми у різних сферах, таких як архітектура,

меблі, ювелірний дизайн та інше. Це дає можливість користувачам налаштувати Rhino 3D під свої потреби та розширювати його функціональність.

Загалом, Rhino 3D є потужним інструментом для тривимірного моделювання і дизайну, який дозволяє створювати складні геометричні форми та поверхні. Він підтримує різні формати файлів і має можливості розширення за допомогою плагінів, що робить його універсальним інструментом для різних галузей дизайну та інженерії.

- **Pro100:**

Pro100 - це програмне забезпечення, що спеціалізується на створенні інтерактивних візуалізацій та проектуванні меблів [15]. Воно надає дизайнерам можливість легко створювати 3D-моделі меблів, докладно розташовувати їх у віртуальних просторах і отримувати живі ілюстрації кінцевого вигляду приміщень.

Програма вражає своєю простотою використання, дозволяючи користувачам вибирати розміри, форму, матеріали та додаткові деталі меблів. Велика бібліотека готових моделей і матеріалів робить процес проектування швидким та зручним.

Окрім того, Pro100 дозволяє генерувати детальні креслення та специфікації для виробництва, що є важливим для передачі технічної інформації виробникам меблів. Таким чином, дизайнери можуть точно передати свої ідеї та вимоги стосовно меблів.

Програма також забезпечує візуалізацію проектів, де користувачі можуть налаштувати освітлення, колір, текстури та інші атрибути, щоб побачити реалістичне зображення меблів інтер'єру.

Загалом, Pro100 є потужним інструментом для дизайну меблів, який пропонує простоту використання, великий вибір моделей і матеріалів, а також можливість створення детальних креслень та візуалізації проектів. Він допомагає дизайнерам ефективно працювати та продемонструвати свої ідеї клієнтам [3].

- **SolidCAM:**

SolidCAM - це інтегрований модуль для автоматизації процесу обробки деталей на станках з програмним керуванням (CNC). Він поєднується з

популярним CAD-системою SolidWorks і надає широкий спектр інструментів для створення програм обробки деталей.

Цей модуль дозволяє дизайнерам і машинобудівним інженерам легко створювати і оптимізувати CNC-програми для різних типів операцій, таких як фрезерування, свердління, токарні операції тощо. Він використовує потужні алгоритми для генерації оптимальних траєкторій руху інструмента, забезпечуючи високу точність та ефективність обробки.

SolidCAM має вбудовані функції автоматичного розпізнавання особливостей геометрії деталей і визначення оптимальних стратегій обробки. Він також підтримує використання різних інструментів і технологій, таких як підпорна система, розширений інструментальний контроль, використання заготовок тощо.

Одна з особливостей SolidCAM - це його інтеграція зі середовищем SolidWorks, що дозволяє користувачам зручно працювати з CAD-моделями і безпосередньо створювати програми обробки на основі геометрії деталей.

Загалом, SolidCAM є потужним інструментом для автоматизації процесу обробки деталей на CNC-станках. Він забезпечує широкий набір інструментів для створення оптимальних програм обробки, має вбудовані функції розпізнавання геометрії і підтримує інтеграцію з CAD-системою SolidWorks. Це допомагає збільшити продуктивність та якість обробки деталей [4].

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1. Мови LISP, AutoLISP та VisualLISP

#### Мова AutoLISP

Мова AutoLISP є діалектом мови програмування LISP (List Processing) і була розроблена для автоматизації завдань у системі автоматизованого проектування AutoCAD. AutoLISP дозволяє розширювати функціональність AutoCAD шляхом написання скриптів і програм, що автоматизують повторювані завдання, створюють нові команди та модифікують об'єкти в кресленні AutoCAD.

AutoLISP базується на синтаксисі LISP, який використовує списки та зворотну польську нотацію. Цей синтаксис дозволяє легко маніпулювати та обробляти дані, що є великою перевагою мови AutoLISP.

Основними рисами мови AutoLISP є:

- **Синтаксис LISP:** AutoLISP використовує синтаксис LISP, що базується на списках і зворотній польській нотації. Синтаксис AutoLISP легкий для вивчення і зрозуміння, що дозволяє швидко створювати програми.

- **Робота з об'єктами AutoCAD:** AutoLISP надає потужні функції для роботи з об'єктами AutoCAD, такими як лінії, кола, блоки тощо. Вона дозволяє зчитувати, створювати, редагувати та аналізувати об'єкти в кресленні AutoCAD.

- **Функціональність:** AutoLISP підтримує рекурсію, умовні вирази, цикли, функції вищого порядку та інші конструкції мови програмування. Це дозволяє розробникам писати потужні та складні програми для виконання різних завдань.

- **Доступ до системних функцій:** AutoLISP може взаємодіяти з системними функціями AutoCAD, такими як команди вводу, редактор тексту та інші. Це дозволяє програмістам виконувати різні операції у середовищі AutoCAD безпосередньо з мови AutoLISP.

- **Розширення функціональності:** AutoLISP дозволяє розширювати функціональність AutoCAD, додавати нові команди та функції, створювати користувацькі інтерфейси та інструменти. Це дає можливість розробникам налаштовувати та адаптувати AutoCAD до своїх потреб і забезпечувати більш ефективний процес проектування.

### **Синтаксис AutoLISP:**

Одним з головних аспектів мови AutoLISP є її синтаксис, який базується на синтаксисі LISP. Основною одиницею програми AutoLISP є вирази, які складаються з списків та атомів. Список є вкладеним набором атомів або інших списків, які визначають функції та аргументи. Атоми можуть бути числами, символами або рядками.

Програми AutoLISP виконуються інтерпретатором, який читає вирази один за одним і виконує відповідні дії. Наприклад, наступний вираз виведе текст "Hello, World!" на екран:

```
(princ "Hello, World!")
```

У цьому виразі princ є функцією, яка виводить значення на екран, а рядок "Hello, World!" є аргументом функції.

AutoLISP також підтримує умовні вирази, цикли та функції вищого порядку.

### **Робота з об'єктами AutoCAD:**

AutoLISP надає потужні функції для роботи з об'єктами AutoCAD. Це означає, що програми AutoLISP можуть зчитувати, створювати, редагувати та аналізувати різні об'єкти, такі як лінії, кола, блоки тощо.

Наприклад, для створення простого кола у AutoCAD за допомогою AutoLISP можна використати наступний код:

```
(command "circle" "center" '(0 0) "radius" 5)
```

У цьому прикладі функція command викликає команду AutoCAD "circle" з необхідними аргументами. Коло буде створено з центром у точці (0 0) та радіусом 5.

AutoLISP також дозволяє отримувати доступ до властивостей об'єктів AutoCAD та змінювати їх

### ***Функціональність AutoLISP:***

AutoLISP підтримує різні функціональні можливості, які дозволяють розробникам писати більш складні та потужні програми.

1. Рекурсія: AutoLISP дозволяє використовувати рекурсію, коли функція викликає сама себе. Це дозволяє ефективно вирішувати завдання, які мають рекурсивну структуру, такі як обробка дерев або списків.
2. Умовні вирази: AutoLISP має умовні вирази, такі як `if`, `cond`, `case`, які дозволяють виконувати різні дії в залежності від умови. Це допомагає зробити програму більш гнучкою та адаптивною.
3. Цикли: AutoLISP підтримує різні види циклів, такі як `repeat`, `foreach`, `while`, `mapcar`. Це дозволяє повторювати певні дії протягом заданої кількості разів або для кожного елемента в списку.
4. Функції вищого порядку: AutoLISP дозволяє використовувати функції вищого порядку, такі як `apply`, `mapcar`, `reduce`. Це дозволяє передавати функції як аргументи і використовувати їх для обробки даних.
5. Робота зі списками: AutoLISP базується на списковій структурі даних і має потужні функції для роботи зі списками. Це дозволяє здійснювати операції, такі як додавання, вилучення, перетворення та обробка списків.

### ***Розширення функціональності AutoCAD:***

Однією з головних переваг мови AutoLISP є її можливість розширювати функціональність системи AutoCAD.

***Додавання нових команд:*** За допомогою AutoLISP можна створювати власні команди, які розширюють функціональність AutoCAD. Це дозволяє створювати короткікоманди для повторюваних дій або для виконання складних обчислень.

**Створення користувацьких інтерфейсів:** AutoLISP дозволяє створювати користувацькі інтерфейси, такі як діалогові вікна, панелі інструментів, меню тощо. Це дає змогу створювати зручні та ефективні інструменти для взаємодії з AutoCAD.

**Редагування креслень:** AutoLISP дозволяє здійснювати редагування об'єктів у кресленнях AutoCAD. Можна змінювати розміри, переміщувати, копіювати, знаходити і замінювати об'єкти в автоматичному режимі.

**Доступ до бази даних:** AutoLISP може взаємодіяти з базою даних AutoCAD і отримувати доступ до відомостей про об'єкти, їх властивості, шари тощо. Це дозволяє автоматизувати процеси аналізу та обробки даних у кресленнях [7].

### **Локальні та глобальні змінні:**

AutoLISP дозволяє використовувати локальні та глобальні змінні для збереження та обміну даними у програмах. Локальні змінні обмежені областю видимості, в якій вони оголошені, і їх значення не доступні ззовні цієї області. Глобальні змінні мають глобальну область видимості і доступні з будь-якого місця програми.

Наприклад, для оголошення та використання локальної змінної можна використовувати функцію `setq`:

```
(defun my-function ()
  (setq x 10)
  (princ x))
```

У цьому прикладі змінна `x` оголошується та ініціалізується значенням 10 всередині функції `my-function`. Значення змінної `x` може бути використане в межах цієї функції.

Для оголошення та використання глобальної змінної можна використати функцію `setq` у глобальному контексті:

```
(setq global-variable 20)
(defun my-function ()
  (princ global-variable))
```

У цьому прикладі змінна `global-variable` оголошується та ініціалізується значенням 20 у глобальному контексті. Значення змінної `global-variable` може бути використане в будь-якій частині програми.

### **Обробка помилок:**

У мові AutoLISP є механізм для обробки помилок, який дозволяє перехоплювати та обробляти непередбачені ситуації під час виконання програм.

Одним із способів обробки помилок є використання конструкції `cond` разом з оператором `error`:

```
(defun my-function (x)
  (cond
    ((not (numberp x))
     (error "Invalid input. Expected a number."))
    (< x 0)
     (error "Invalid input. Number must be non-negative."))
    (t
     (princ "Valid input.")))))
```

У цьому прикладі функція `my-function` перевіряє, чи вхідний параметр `x` є числом і чи є він невід'ємним. Якщо перевірка не пройдена, генерується помилка з відповідним повідомленням. Якщо перевірка пройдена, виводиться повідомлення про коректний ввід.

Також, можна використовувати конструкцію `try/catch` для перехоплення та обробки помилок:

```
(defun my-function (x)
  (try
    (if (not (numberp x))
        (throw 'error "Invalid input. Expected a number."))
    (if (< x 0)
```

```
(throw 'error "Invalid input. Number must be non-
negative."))
(princ "Valid input.")
(catch 'error (message) (princ message)))
```

У цьому прикладі функція `my-function` використовує конструкцію `try/catch` для перехоплення помилки. Якщо виникає помилка, вона перехоплюється та повідомлення помилки виводиться на екран.

Обробка помилок в AutoLISP дозволяє покращити надійність та стабільність програм, забезпечуючи контроль над непередбаченими ситуаціями та відповідну реакцію на них.

### ***Робота з файлами:***

Мова AutoLISP також надає можливість працювати з файлами, що відкриває широкі можливості для обробки та обміну даними.

- Відкриття та закриття файлів: За допомогою функцій `open` та `close` можна відкривати та закривати файли.
- Зчитування та запис даних: AutoLISP має різні функції для зчитування та запису даних у файл. Наприклад, функція `read-line` дозволяє зчитувати один рядок з файлу, а функція `write-line` дозволяє записувати рядок у файл.
- Переміщення по файлу: AutoLISP надає функції для переміщення покажчика у файлі. Наприклад, функція `seek` дозволяє встановити покажчик на певну позицію у файлі.
- Робота з каталогами: AutoLISP має функції для роботи з каталогами, такі як `setq` та `directory`. Наприклад, функція `setq` дозволяє встановити поточний каталог, а функція `directory` дозволяє отримати список файлів та каталогів у заданому каталозі [7].

## **Використання AutoLISP у практичних сценаріях:**

Мова AutoLISP знайшла широке застосування у практичних сценаріях, пов'язаних з проектуванням та моделюванням. Ось кілька прикладів використання AutoLISP:

1. Автоматизація рутинних задач: AutoLISP дозволяє автоматизувати рутинні задачі у системі AutoCAD, такі як створення повторюваних об'єктів, зміна властивостей об'єктів, генерація звітів тощо. Це допомагає зберегти час та зусилля проектувальників.
2. Розробка налаштованих інструментів: AutoLISP дозволяє створювати власні налаштовані інструменти у системі AutoCAD. Наприклад, можна розробити власну панель інструментів з необхідними командами, або створити спеціалізовані функції для конкретного проекту.
3. Аналіз та обробка даних: AutoLISP дозволяє взаємодіяти з базою даних AutoCAD та проводити аналіз та обробку даних у кресленнях. Наприклад, можна знаходити та підраховувати кількість певних об'єктів, фільтрувати об'єкти за певними критеріями, тощо.
4. Розширення функціональності AutoCAD: AutoLISP дозволяє розширити функціональність системи AutoCAD шляхом створення власних команд та функцій. Наприклад, можна розробити власний алгоритм обробки об'єктів або створити власні діалогові вікна для взаємодії з користувачем.

Мова AutoLISP є потужним інструментом для автоматизації та розширення функціональності системи AutoCAD. Вона надає зручні засоби для роботи з об'єктами AutoCAD, базою даних, файлами та дозволяє виконувати різноманітні завдання пов'язані з проектуванням та моделюванням. Знання мови AutoLISP відкриває безліч можливостей для розробки налаштованих інструментів та автоматизації процесів у AutoCAD.

## Мова Visual LISP

Visual LISP (VLISP) є мовою програмування, яка використовується для розширення функціональності програмного забезпечення AutoCAD, відомої системи комп'ютерного проектування і черчення. Вона надає розширені можливості для автоматизації та налаштування процесів, що дозволяє користувачам створювати спеціалізовані програми, інструменти та функції, які допомагають збільшити продуктивність і зручність роботи з AutoCAD.

Visual LISP базується на мові програмування AutoLISP, але надає додаткові можливості та функції, які полегшують розробку і підтримку програм, що використовуються в AutoCAD. Вона надає доступ до об'єктної моделі AutoCAD, дозволяючи програмістам працювати з графічними об'єктами, такими як лінії, кола, текст, блоки тощо, та виконувати з ними різноманітні операції.

Visual LISP також має багатий набір вбудованих функцій та операцій, що дозволяють маніпулювати даними, здійснювати обчислення, працювати зі списками та рядками, керувати потоком виконання програми за допомогою умовних виразів та циклів. Крім того, Visual LISP підтримує можливість створення власних функцій, що дозволяє програмістам групувати код та використовувати його у багатьох місцях програми, що спрощує розробку та підтримку складних проектів.

Однією з головних переваг Visual LISP є те, що вона інтегрована безпосередньо в середовище AutoCAD, що спрощує процес розробки та тестування програм. Програми на Visual LISP можуть бути виконані безпосередньо в середовищі AutoCAD, що дозволяє програмістам швидко перевірити та відлагодити свій код.

Загалом, Visual LISP є потужним інструментом для розширення можливостей AutoCAD та автоматизації рутинних завдань черчення та моделювання, що дозволяє програмістам створювати спеціалізовані програми та налаштовувати робоче оточення AutoCAD відповідно до своїх потреб.

Visual LISP надає додаткові можливості та функції, які полегшують розробку та підтримку програм, що використовуються в AutoCAD. Ось деякі з цих можливостей:

1. Доступ до об'єктної моделі AutoCAD: Visual LISP дозволяє отримувати доступ до різних об'єктів у чертежі, таких як точки, лінії, кола, текст, блоки тощо. Це відкриває широкі можливості для маніпуляції цими об'єктами, зміни їх властивостей, створення нових об'єктів та виконання різноманітних операцій з ними.
2. Робота зі списками та рядками: Visual LISP надає багато функцій для роботи зі списками та рядками даних. Це дозволяє легко виконувати операції з даними, такі як додавання, видалення, зміна, пошук тощо. Можна легко маніпулювати списками та рядками, що спрощує обробку даних в програмах AutoCAD.
3. Можливість створення власних функцій: Visual LISP дозволяє створювати власні функції, що спрощує організацію коду та використання його у багатьох частинах програми. Власні функції можуть приймати параметри, виконувати певні дії та повертати значення. Це дозволяє програмістам створювати власні набори функцій, які відповідають їх потребам і спрощують розробку складних програм.
4. Механізми обробки помилок: Visual LISP надає можливості для обробки помилок та винятків. Це дозволяє контролювати виконання програми та реагувати на непередбачені ситуації. Можна генерувати винятки, визначати власні обробники помилок та виконувати певні дії при виникненні помилки.
5. Інтеграція з іншими мовами програмування: Visual LISP може взаємодіяти з іншими мовами програмування, такими як VBA (Visual Basic for Applications) або .NET Framework. Це дозволяє розробникам

комбінувати різні мови програмування для створення потужних та розширених програм в середовищі AutoCAD.

Visual LISP надає широкі можливості для розробки програм в AutoCAD, дозволяючи програмістам створювати спеціалізовані інструменти, автоматизувати рутинні задачі та забезпечувати більш зручне та продуктивне робоче оточення.

Visual LISP має велику кількість вбудованих функцій та операцій, які роблять розробку програм в середовищі AutoCAD більш зручною та продуктивною. Завдяки цим функціям, розробники можуть легко працювати з математичними операціями, маніпулювати списками та рядками, обробляти умовні вирази та цикли, працювати з файлами та керувати блоками та атрибутами в чертежах. Цей розширений набір можливостей дозволяє створювати складні та ефективні програми, які автоматизують процеси черчення та моделювання в AutoCAD.

#### Цікаві факти про мову AutoLISP

1. Історія: Visual LISP була розроблена компанією Autodesk для використання в середовищі AutoCAD. Перша версія Visual LISP була представлена у 1997 році, і вона постійно покращується та оновлюється разом з новими версіями AutoCAD.
2. Інтеграція з AutoCAD: Visual LISP є однією з найпотужніших мов програмування, які вбудовані безпосередньо в середовище AutoCAD. Це дозволяє розробникам безпосередньо взаємодіяти з об'єктами чертежів, блоками, атрибутами та іншими компонентами AutoCAD, що робить його дуже потужним для автоматизації черчення та моделювання.
3. Функції розширення: Visual LISP надає можливості розширення середовища AutoCAD. Завдяки Visual LISP, розробники можуть створювати спеціалізовані інструменти, додавати нові команди, меню

та панелі інструментів, налаштовувати робочий простір AutoCAD та реалізовувати інші розширення, які відповідають їх потребам.

4. Інтеграція з іншими мовами програмування: Visual LISP може взаємодіяти з іншими мовами програмування, такими як VBA (Visual Basic for Applications) або .NET Framework. Це дає розробникам можливість комбінувати різні мови програмування та використовувати їх у спільних проектах.
5. Широке співтовариство користувачів: Visual LISP має велике та активне співтовариство користувачів, що сприяє обміну знаннями, дослідженню нових технік та розробці високоякісних програм. Існує багато ресурсів, таких як форуми, блоги, вебінари та онлайн-курси, які допомагають розробникам вивчати та покращувати свої навички у використанні Visual LISP.
6. Підтримка стандартів: Visual LISP підтримує стандарти програмування, такі як Common Lisp та ANSI AutoLISP. Це забезпечує сумісність та переносимість коду між різними системами, а також сприяє розвитку добрих практик програмування.
7. Потужність та гнучкість: Visual LISP надає розробникам велику потужність та гнучкість для створення складних програм з великою кількістю функцій та можливостей. Завдяки цьому, Visual LISP використовується для створення різноманітних рішень, від простих автоматизованих скриптів до складних програмних розширень для AutoCAD [5].

## **2.2. Засоби інформаційного забезпечення**

Засоби інформаційного забезпечення є необхідними для ефективної роботи організацій та забезпечення їх інформаційних потреб. Ці засоби

включають інформаційні системи, комп'ютерне обладнання та програмне забезпечення.

Засоби інформаційного забезпечення - це комплекс технологій, програм, апаратного та програмного забезпечення, які використовуються для збору, обробки, зберігання та передачі інформації. Вони включають різноманітні інформаційні системи, комп'ютерне обладнання, програмне забезпечення та мережі зв'язку. Засоби інформаційного забезпечення використовуються для автоматизації бізнес-процесів, забезпечення доступу до інформації, оптимізації робочих процесів та підвищення продуктивності організацій.

Засоби інформаційного забезпечення відіграють важливу роль у сучасному світі, де інформація є ключовим ресурсом для багатьох сфер діяльності. Основні аспекти важливості засобів інформаційного забезпечення включають:

1. Покращення ефективності та продуктивності: Засоби інформаційного забезпечення дозволяють автоматизувати бізнес-процеси, спрощуючи рутинні операції та забезпечуючи швидкий доступ до інформації. Це дозволяє підвищити ефективність та продуктивність роботи організацій.

2. Збереження та захист інформації: Засоби інформаційного забезпечення забезпечують збереження та захист важливої інформації. Вони включають системи з резервним копіюванням даних, системи шифрування та захисту від несанкціонованого доступу, що допомагає уникнути втрати чи пошкодження даних.

3. Покращення прийняття рішень: Засоби інформаційного забезпечення надають організаціям доступ до потрібної інформації для прийняття обґрунтованих рішень. Вони забезпечують аналітичні засоби, системи управління базами даних та інші інструменти, які допомагають здійснювати аналіз інформації та прогнозування.

4. Покращення комунікації та співпраці: Засоби інформаційного забезпечення сприяють покращенню комунікації та співпраці в організаціях.

Вони дозволяють обмінюватись даними, спілкуватись та співпрацювати в режимі реального часу, незалежно від фізичного розташування.

5. Підтримка стратегічного призначення: Засоби інформаційного забезпечення є важливими для реалізації стратегічних цілей організацій. Вони допомагають збирати та аналізувати дані, здійснювати прогнозування, а також підтримують процеси планування та контролю.

Засоби інформаційного забезпечення є необхідними для ефективного функціонування організацій та забезпечення їх інформаційних потреб. Вони дозволяють автоматизувати процеси, покращити прийняття рішень, забезпечити збереження та захист інформації, покращити комунікацію та співпрацю, а також підтримувати стратегічне призначення організацій. Розуміння та використання цих засобів є ключовим для успіху в сучасному інформаційному середовищі.

Інформаційна система (ІС) - це комплекс взаємопов'язаних компонентів, які співпрацюють для збору, обробки, зберігання та передачі інформації з метою підтримки прийняття рішень, виконання операцій та досягнення цілей організації. Інформаційні системи можуть використовуватись у різних сферах, таких як бізнес, освіта, наука, медицина тощо.

Інформаційна система складається з різних компонентів, які взаємодіють між собою для забезпечення її функціонування. Основні складові частини ІС включають:

1. Апаратне та програмне забезпечення: Це включає комп'ютери, сервери, мережеве обладнання та програмне забезпечення, необхідне для роботи ІС. Апаратне забезпечення забезпечує фізичну інфраструктуру, а програмне забезпечення - функціональні можливості, такі як операційні системи, програми для обробки даних, системи управління базами даних тощо.

2. Бази даних: Це структуровані збірки даних, які зберігаються та організовуються в ІС. Бази даних забезпечують зручний доступ до інформації та дозволяють виконувати операції збереження, оновлення, пошуку та аналізу даних.

3. Мережі зв'язку: Це інфраструктура, що дозволяє передавати дані між різними компонентами ІС. Мережі зв'язку можуть бути локальними (LAN), глобальними (WAN), бездротовими (Wi-Fi) тощо. Вони забезпечують швидку та безпечну передачу даних.

4. Користувачі: Це люди, які взаємодіють з ІС для отримання, внесення або обробки інформації. Користувачі включають адміністраторів, операторів, аналітиків та інших фахівців, які використовують систему для своєї роботи.

Побудова та функціонування інформаційних систем ґрунтується на кількох принципах:

1. Системний підхід: Інформаційна система розглядається як єдина система, яка включає в себе взаємопов'язані компоненти. Цей підхід дозволяє розглядати систему в цілому та визначати взаємозв'язки та вплив компонентів один на одного.

2. Життєвий цикл: Інформаційні системи проходять через різні етапи свого життєвого циклу, включаючи аналіз вимог, проектування, розробку, впровадження та підтримку. Кожен етап вимагає виконання певних задач та врахування специфіки системи.

3. Гнучкість та масштабованість: Інформаційні системи повинні бути гнучкими та легко масштабовуватися, щоб відповідати змінним потребам організацій. Вони повинні дозволяти внесення змін, розширення функціональності та підтримку зростаючого обсягу даних та користувачів.

4. Забезпечення безпеки: Інформаційні системи повинні бути захищеними від несанкціонованого доступу, втрати чи пошкодження даних. Забезпечення безпеки включає використання методів шифрування, аутентифікації, контролю доступу та резервного копіювання даних.

5. Постійна підтримка та оновлення: Інформаційні системи потребують постійної підтримки, включаючи виявлення та виправлення помилок, оновлення програмного та апаратного забезпечення, а також надання користувачам необхідної підтримки та навчання.

Інформаційні системи є складними комплексами, що включають апаратне та програмне забезпечення, бази даних, мережі зв'язку та користувачів. Вони побудовані на принципах системного підходу, життєвого циклу, гнучкості та масштабованості, забезпечення безпеки та постійної підтримки. Розуміння цих принципів допомагає ефективно використовувати інформаційні системи для досягнення цілей організації та забезпечення її успіху.

#### Класифікація інформаційних систем

Транзакційні системи є одним з основних типів інформаційних систем і використовуються для збереження та обробки операційних даних. Вони спрямовані на автоматизацію та оптимізацію операційних процесів, зокрема обробку транзакцій, які включають покупку, продаж, оплату тощо. Транзакційні системи забезпечують точність, надійність та швидкість обробки великого обсягу операційних даних.

Управлінські інформаційні системи призначені для підтримки управлінського процесу в організації. Вони забезпечують збір, аналіз та представлення даних, необхідних для прийняття рішень на різних рівнях управління. Управлінські інформаційні системи можуть включати звітність, аналітичні засоби, планування ресурсів, системи управління взаємодією з клієнтами та інші компоненти для підтримки управлінських функцій.

Експертні системи - це інформаційні системи, які забезпечують експертні знання та досвід в певній предметній області. Вони використовуються для прийняття рішень на основі логіки та правил, що визначені експертами відповідної галузі. Експертні системи можуть бути використані у медицині, фінансах, інженерії та інших сферах, де потрібна висока експертиза [6].

Окрім вищезгаданих типів інформаційних систем, існує багато інших типів, таких як системи планування ресурсів підприємства (Enterprise Resource Planning - ERP), системи управління відносинами з клієнтами (Customer Relationship Management - CRM), системи управління проектами, системи управління ланцюгом постачання та багато інших. Кожен з цих типів

інформаційних систем має свої функції та можливості, спеціально розроблені для конкретних потреб організації чи галузі.

Опис функцій та можливостей кожного типу інформаційної системи

1. Транзакційні системи:

- Збереження та обробка операційних даних, зокрема транзакцій, покупок, продажів, оплат тощо.
- Автоматизація та оптимізація операційних процесів.
- Забезпечення точності, надійності та швидкості обробки великого обсягу операційних даних.
- Ведення журналу транзакцій та забезпечення цілісності даних.

2. Управлінські інформаційні системи:

- Збір, аналіз та представлення даних для прийняття управлінських рішень.
- Підтримка планування ресурсів, аналітики та звітності.
- Визначення ключових показників ефективності та моніторинг їх досягнення.
- Управління взаємодією з клієнтами та сторонніми партнерами.

3. Експертні системи:

- Забезпечення експертних знань та досвіду в певній предметній області.
- Використання логіки та правил для прийняття рішень на основі експертних знань.
- Діагностика та розв'язання проблем на основі експертної експертизи.
- Підтримка прийняття рішень у галузях, де висока експертиза є важливою (медицина, фінанси, інженерія тощо).

4. Системи підтримки прийняття рішень та інші типи:

- Забезпечення підтримки прийняття рішень на основі аналізу даних та моделювання ситуацій.

- Використання статистичних методів та алгоритмів для прогнозування та оптимізації.
- Підтримка прийняття стратегічних рішень на основі сценаріїв та симуляцій.
- Розробка рекомендацій та альтернативних варіантів дій [6].

Кожен тип інформаційної системи має свої специфічні функції та можливості, які відповідають конкретним потребам організації чи галузі. Використання відповідної інформаційної системи допомагає підтримувати ефективність роботи, поліпшувати прийняття рішень та досягати стратегічних цілей організації.

Засоби інформаційного забезпечення, такі як інформаційні системи, відіграють важливу роль у забезпеченні ефективності, продуктивності та конкурентоспроможності організацій.

Інформаційні системи є складними комплексами, які включають апаратне та програмне забезпечення, бази даних, мережі зв'язку та користувачів. Вони побудовані на принципах системного підходу, життєвого циклу, гнучкості та масштабованості, забезпечення безпеки та постійної підтримки. Розуміння цих принципів допомагає ефективно використовувати інформаційні системи для досягнення цілей організації та забезпечення її успіху.

Класифікація інформаційних систем розкриває різні типи систем, такі як транзакційні системи, управлінські інформаційні системи, експертні системи, системи підтримки прийняття рішень та інші. Кожен тип системи має свої функції та можливості, спеціально розроблені для конкретних потреб організації чи галузі.

Засоби інформаційного забезпечення є необхідними для забезпечення ефективної обробки та управління інформацією, забезпечення доступу до неї, прийняття раціональних рішень та покращення ділових процесів. Вони допомагають підвищити ефективність виробництва, знизити витрати, поліпшити

якість продукції чи послуг і підвищити конкурентоспроможність організації на ринку.

Таким чином, розуміння і використання засобів інформаційного забезпечення стає необхідним у сучасному бізнес-середовищі. Їх ефективне використання допомагає організаціям досягати стратегічних цілей, підвищувати продуктивність та конкурентоспроможність, а також пристосовуватися до швидкозмінних умов ринку.

### **2.3. Засоби математичного забезпечення**

"Засоби математичного забезпечення" мають велику значимість у сучасному світі, де великі обсяги даних та складні завдання вимагають аналізу та оптимізації. Використання математичних методів та моделей дозволяє ефективно обробляти та аналізувати інформацію, здійснювати прогнозування та приймати обґрунтовані рішення. Ми розглянемо загальні математичні методи, їх застосування у різних сферах діяльності, а також обговоримо переваги та обмеження використання цих методів.

#### Математичні методи та моделі

У сучасному інформаційному забезпеченні широко застосовуються різні математичні методи та моделі, які допомагають здійснювати аналіз даних, розв'язувати оптимізаційні задачі та прогнозувати результати. Деякі загальні математичні методи, які використовуються в інформаційному забезпеченні, включають:

1. Лінійне програмування: Це метод оптимізації, який дозволяє знайти найкращий розв'язок задачі при врахуванні обмежень та цільової функції.
2. Теорія ймовірностей: Цей математичний підхід дозволяє моделювати та аналізувати невизначеність та ризики в процесі прийняття рішень.
3. Дискретне моделювання: Використовується для аналізу систем з обмеженим набором можливих станів та подій.

4. Числові методи: Включають чисельне інтегрування, диференціювання, розв'язування диференціальних рівнянь та інші чисельні алгоритми.
5. Графова теорія: Використовується для моделювання та аналізу зв'язків між об'єктами та подіями у формі графів.

Приклади застосування математичних методів у різних сферах діяльності вражають своєю розмаїтістю. Наприклад, в економіці математичні методи використовуються для прогнозування ринкових тенденцій та моделювання фінансових ризиків. У медицині математичні моделі допомагають виявляти залежності між симптомами та хворобами, покращувати діагностику та прогнозувати результати лікування. В транспортній сфері математичні методи застосовуються для оптимізації маршрутів транспортних засобів та управління трафіком.

Необхідно також обговорити переваги та обмеження використання математичних методів. До переваг можна віднести точність та обґрунтованість результатів, можливість швидкого аналізу великих обсягів даних та автоматизацію процесів. Однак, математичні методи мають свої обмеження, зокрема, вони базуються на припущеннях та спрощеннях, що можуть призвести до неточностей у моделюванні реальних процесів.

Статистичний аналіз є важливою складовою засобів математичного забезпечення, використовуваних у сфері інформаційного забезпечення.

Опис статистичних методів та підходів

Статистичний аналіз базується на математичних та ймовірнісних принципах і використовується для збору, опису, аналізу та інтерпретації даних. Деякі з основних статистичних методів включають:

1. Дескриптивна статистика: Цей підхід використовується для опису та визначення основних характеристик даних, таких як середнє значення, медіана, дисперсія та інші.
2. Інференційна статистика: Цей метод використовується для зроблення висновків та припущень про популяцію на основі обмеженого набору

даних (вибірки). Включає оцінку параметрів, перевірку гіпотез, інтервальну оцінку та інші методи.

3. Кореляційний аналіз: Цей метод досліджує залежності між різними змінними та визначає силу та напрямок цих залежностей.
4. Регресійний аналіз: Цей метод дозволяє встановити функціональну залежність між залежною та незалежними змінними та провести прогнози.

Приклади застосування статистичного аналізу

Статистичний аналіз застосовується у багатьох сферах діяльності для виявлення закономірностей, розуміння трендів та прийняття обґрунтованих рішень. Ось кілька прикладів:

1. В маркетингу: Статистичний аналіз допомагає визначити профіль цільової аудиторії, прогнозувати попит на товари та послуги, оцінювати ефективність маркетингових кампаній.
2. В фінансах: Статистичний аналіз дозволяє моделювати фінансові ризики, прогнозувати ціни акцій та валют, оцінювати доходність портфелю та приймати рішення щодо інвестицій.
3. В медицині: Статистичний аналіз допомагає вивчати ефективність лікування, виявляти фактори ризику для захворювань, проводити клінічні дослідження та приймати рішення щодо здоров'я населення.

Значення статистичного аналізу у прийнятті стратегічних рішень

Статистичний аналіз грає важливу роль у прийнятті стратегічних рішень в організаціях. Він надає можливість об'єктивно оцінювати ситуацію, виявляти тенденції та розуміти взаємозв'язки між різними факторами. Це дозволяє керівникам та менеджерам приймати обґрунтовані рішення, зменшувати ризики та забезпечувати успішний розвиток організації [7].

Математичне програмування є потужним інструментом, який використовує математичні моделі та алгоритми для розв'язування оптимізаційних задач. В цьому розділі ми розглянемо поняття та принципи

математичного програмування, основні типи оптимізаційних задач і приклади їх застосування для оптимізації процесів та прийняття рішень.

Огляд поняття та принципів математичного програмування

Математичне програмування - це галузь оптимізації, яка займається знаходженням найкращого рішення (найбільшого або найменшого значення) серед обмежень та обмежень нашої системи. Основні принципи математичного програмування включають:

1. Цільова функція: Визначення цілі, яку ми намагаємося максимізувати або мінімізувати. Це може бути, наприклад, максимізація прибутку або мінімізація витрат.
2. Обмеження: Встановлення обмежень або умов, які повинні виконуватись у розв'язку. Це можуть бути обмеження ресурсів, технічні обмеження або правові обмеження.
3. Математичні моделі: Представлення проблеми у формі математичних рівнянь та нерівностей, що описують взаємозв'язки між змінними.
4. Методи розв'язання: Використання алгоритмів та методів для знаходження оптимального рішення, таких як симплекс-метод, метод градієнта, генетичні алгоритми та інші.

Основні типи оптимізаційних задач

У математичному програмуванні існує кілька основних типів оптимізаційних задач, зокрема:

1. Лінійне програмування: Задачі, в яких цільова функція та обмеження є лінійними функціями.
2. Цілочисельне програмування: Задачі, в яких значення змінних обмежені цілими числами.
3. Нелінійне програмування: Задачі, в яких цільова функція або обмеження є нелінійними функціями.
4. Динамічне програмування: Задачі, в яких розв'язок залежить від послідовного вибору рішень протягом певного часового періоду.

5. Мультикритеріальне програмування: Задачі, в яких цільова функція має кілька критеріїв, що потрібно врахувати [8].

Приклади застосування математичного програмування

Математичне програмування застосовується в різних сферах для вирішення різноманітних проблем. Ось кілька прикладів:

1. Логістика: Математичне програмування використовується для оптимізації маршрутів транспорту, планування поставок та складського управління.
2. Виробництво: Використання математичного програмування допомагає встановити оптимальні розклади виробництва, розподіл ресурсів та планування виробничих процесів.
3. Фінанси: Математичне програмування використовується для оптимізації інвестиційних портфелів, управління ризиками та розрахунку оптимальних стратегій фінансових операцій.
4. Транспорт: Використання математичного програмування допомагає планувати маршрути громадського транспорту, розміщення транспортних вузлів та оптимізацію транспортної інфраструктури.

Результати використання математичного програмування в цих і багатьох інших сферах показують його значення для оптимізації процесів, підвищення ефективності та прийняття обґрунтованих рішень.

Системи символного обчислення (ССО) є потужними інструментами, які дозволяють виконувати обчислення з використанням символів та алгебраїчних виразів замість чисел. Основна ідея ССО полягає в тому, що вони працюють з виразами у символній формі, зберігаючи всі символні властивості та спрощуючи вирази згідно з математичними правилами.

Основні принципи та можливості систем символного обчислення включають:

1. Маніпуляції з символами: ССО дозволяють виконувати операції з символами, такі як спрощення, розкладання, факторизація, пошук

спільних членів тощо. Це дозволяє виконувати складні алгебраїчні маніпуляції та отримувати точні символічні результати.

2. Вирішення рівнянь та систем рівнянь: ССО можуть вирішувати різні типи рівнянь, включаючи лінійні, квадратичні, трансцендентні тощо. Вони також здатні вирішувати системи рівнянь, що дозволяє знаходити значення невідомих змінних.
3. Диференціювання та інтегрування: ССО здатні диференціювати та інтегрувати символічні вирази, надаючи точні аналітичні похідні та неозначені та визначені інтеграли. Це дозволяє проводити математичний аналіз та розв'язувати задачі з калькулюсу.
4. Символьна алгебра: ССО надають широкі можливості для роботи з символічними виразами, включаючи алгебраїчні маніпуляції, спрощення, факторизацію, розкладання на множники, підстановку значень змінних тощо. Це дозволяє ефективно виконувати алгебраїчні операції та отримувати точні символічні результати.
5. Символьні обчислення в фізиці та інженерії: ССО використовуються в фізиці та інженерії для моделювання фізичних систем, розв'язування складних математичних рівнянь, виконання обчислень та аналізу результатів. Вони дозволяють проводити символічний аналіз та оптимізувати процеси проектування та розробки.

Приклади використання систем символічного обчислення включають:

- Розв'язання складних алгебраїчних та трансцендентних рівнянь.
- Обчислення точних похідних та інтегралів.
- Аналітичне моделювання фізичних систем та процесів.
- Вирішення задач оптимізації та пошуку екстремумів.
- Аналіз та побудова математичних моделей у наукових та інженерних дослідженнях.

Застосування ССО дозволяє ефективно працювати з символічними виразами, отримувати точні символічні результати та виконувати складні

математичні обчислення. Це забезпечує широкі можливості для аналізу даних, моделювання систем та прийняття обґрунтованих рішень.

Засоби математичного забезпечення, такі як математичні методи та моделі, статистичний аналіз, математичне програмування та системи символного обчислення, відіграють важливу роль у розв'язанні складних завдань та прийнятті обґрунтованих рішень.

Математичні методи та моделі дозволяють аналізувати дані, прогнозувати тенденції, виявляти закономірності та робити розрахунки. Вони забезпечують точність та об'єктивність в аналізі інформації, що допомагає приймати обґрунтовані рішення.

Статистичний аналіз дозволяє виявляти кореляції, залежності та розподіли в даних. Він є потужним інструментом для виявлення закономірностей та вирішення проблем у багатьох сферах діяльності, включаючи маркетингові дослідження, фінансовий аналіз, якість контролю тощо.

Математичне програмування надає можливість розв'язувати складні оптимізаційні задачі та вирішувати проблеми з ресурсним плануванням, транспортними потоками, логістикою та багатьма іншими. Воно допомагає знайти оптимальні рішення та знизити витрати.

Системи символного обчислення дозволяють працювати з символними виразами, проводити алгебраїчні маніпуляції та розв'язувати складні математичні вирази. Вони знаходять застосування у різних галузях, таких як фізика, інженерія, математика, де важливо мати точні символні результати.

Засоби математичного забезпечення відіграють критичну роль у сфері інформаційного забезпечення. Вони допомагають аналізувати великі обсяги даних, виявляти тенденції та залежності, розв'язувати складні задачі оптимізації та приймати обґрунтовані рішення. Завдяки їм організації можуть покращити свою продуктивність, ефективність та конкурентоспроможність.

Таким чином, використання математичних засобів у контексті інформаційного забезпечення є необхідним та корисним для організацій та

сфери діяльності. Вони допомагають зрозуміти дані, знайти закономірності та розв'язати складні проблеми. Інтеграція цих засобів у процеси прийняття рішень сприяє покращенню стратегічного планування, оптимізації ресурсів та досягненню успіху в сучасному інформаційному середовищі [9].

## РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1. Основні відомості про AutoCAD

AutoCAD (АвтоКАД) є одним з найпопулярніших програмних продуктів у сфері комп'ютерного проектування та креслення. Вперше випущений в 1982 році, AutoCAD став стандартом для інженерів, архітекторів та дизайнерів.

Основні відомості про AutoCAD:

1. Функції та можливості: AutoCAD дозволяє створювати, редагувати та переглядати 2D та 3D креслення та моделі. Він надає широкі можливості для роботи з геометричними об'єктами, текстом, розмірами, шарами та іншими елементами проекту. AutoCAD також має вбудовані інструменти для розрахунку та аналізу параметрів конструкцій, візуалізації проектів та підготовки документації.
2. Інтерфейс: AutoCAD має зручний інтуїтивний інтерфейс, що дозволяє користувачам швидко орієнтуватися в програмі та виконувати різноманітні завдання. Він має меню, панелі інструментів, командний рядок та інші елементи для зручного доступу до функцій та налаштувань.
3. Підтримка форматів файлів: AutoCAD підтримує різноманітні формати файлів, включаючи DWG (основний формат AutoCAD), DXF (формат для обміну даними), PDF (для збереження та обміну документацією) та інші. Це дає можливість легко обмінюватися проектами з іншими користувачами та програмами.
4. Розширення та налаштування: AutoCAD підтримує розширення та налаштування, що дозволяє користувачам розширити функціональність програми за допомогою плагінів, додаткових модулів та налаштувань. Це дозволяє адаптувати AutoCAD до конкретних потреб та вимог користувача.

5. Використання в різних галузях: AutoCAD застосовується в багатьох галузях, включаючи архітектуру, інженерію, будівництво, машинобудування, геодезію та інші. Він дозволяє фахівцям створювати детальні креслення, 3D-моделі та технічну документацію для проектів будь-якої складності.

AutoCAD є потужним інструментом для проектування та креслення, що надає широкі можливості для творчого вияву та реалізації ідей у сфері дизайну та інженерії. Його широке використання і приваблива функціональність роблять його незамінним засобом для професіоналів у багатьох галузях [10].

### **3.2. Реалізація програмного забезпечення**

Проектування меблів для передпокою реалізовано на прикладі передпокою “Грейд Комфорт”, рис.3.1 та рис.3.3.

Дане програмне забезпечення працює у вигляді додатку для AutoCAD2023 і реалізує побудову передпокою, рис.3.2. Додаток написаний мовами програмування AutoLISP, VisualLisp та DCL у середовищі VisualLISP for AutoCAD.

Відповідно до вимог із технічного завдання було передбачено виведення усіх результатів геометричних розрахунків у файл для подальшого створення необхідної документації, рис.3.4 та рис.3.5.

Під час виводу геометричних розрахунків у файл було враховано конструктивні особливості кожного елемента і виведено крім загальних чистові розміри, без личківки, кожного елемента передпокою.

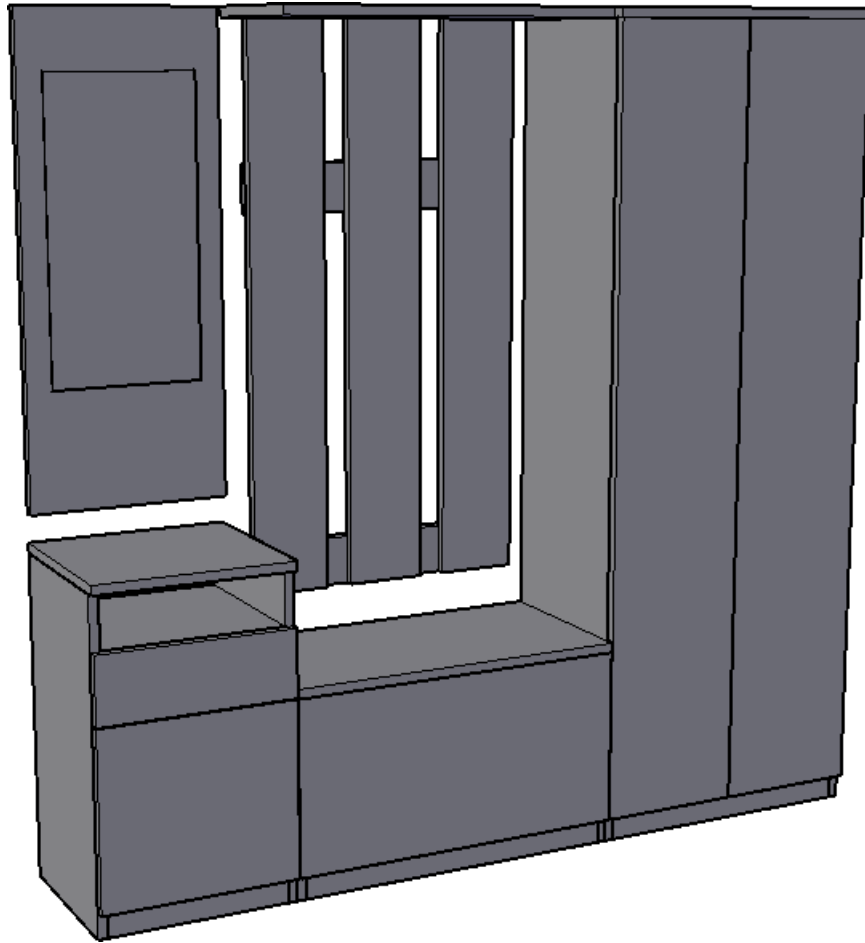


Рис.3.1. Концептуальний вигляд (в перспективі) передпокою у середовищі AutoCAD2023

Додаток AutoCAD для побудови передпокою 'Грейд Комфорт', каф. ІСКМ НЛТУ України

глибина передпокою (1):	370.0	висота цоколя (10):	50.0
ширина 1 секції із дзеркалом (2):	402.0	висота шухляди (11):	140.0
ширина 2 секції із вішаками (3):	700.0	висота полки у шафі (12):	340.0
ширина 3 секції із шафою (4):	600.0	розмір технологічного зазору між шухлядами (13):	2.0
висота передпокою (5):	1700.0	товщина плити	18мм
висота тумби під дзеркалом (6):	680.0	товщина облицювальної кромки	2мм
висота тумби під вішаками (7):	430.0	шлях для збереження файлу	C:/greйд/1.txt
розмір горизонтальної планки вішаків (8):	97.0	<input checked="" type="checkbox"/> зберегти результати у текстовий файл	
розмір вертикальної планки вішаків (9):	160.0		

OK Cancel

Рис.3.2. Діалогове вікно побудови передпокою



Рис. 3.3. Тонований передпокій у середовищі AutoCAD2023

1.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Розміри заготовок плити з врахуванням личківки:

Розміри заготовок передпокою:

1. Розмір лівої бокової стінки тумби, що знаходиться під дзеркалом:----- (352.0 662.0)
2. Розмір правої бокової стінки тумби, що знаходиться під дзеркалом:----- (352.0 662.0)
3. Розмір лівої бокової стінки тумби під вішаками:----- (352.0 412.0)
4. Розмір правої бокової стінки тумби під вішаками:----- (352.0 412.0)
5. Розмір правої бокової стінки шафи:----- (352.0 1932.0)
6. Розмір лівої бокової стінки шафи:----- (352.0 1932.0)
7. Розмір верха тумби, що знаходиться під дзеркалом:----- (370.0 502.0)
8. Розмір верха тумби, що знаходиться під дзеркалом:----- (370.0 502.0)
9. Розмір низа тумби, що знаходиться під дзеркалом:----- (352.0 466.0)
10. Розмір верха тумби, що знаходиться під вішаками:----- (370.0 700.0)
11. Розмір низа тумби, що знаходиться під вішаками:----- (352.0 664.0)
12. Розмір горизонтального верха вішаків:----- (340.0 700.0)
13. Розмір верха шафи:----- (370.0 600.0)
14. Розмір верхньої полицки у шафі:----- (352.0 564.0)
15. Розмір низа шафи:----- (352.0 564.0)
16. Розмір полицки над шухлядою:----- (352.0 466.0)
17. Розмір дверки тумби, що знаходиться під дзеркалом:----- (498.0 358.0)
18. Розмір передньої лицевої стінки шухляди:----- (498.0 136.0)
19. Розмір дверки тумби, що знаходиться під вішаками:----- (696.0 358.0)
20. Розмір лівої дверки шафи:----- (296.0 1878.0)
21. Розмір правої дверки шафи:----- (296.0 1878.0)
22. Розмір горизонтальної планки вішаків:-----2шт.----- (590.0 97.0)
23. Розмір вертикальної планки вішаків:-----3шт.----- (160.0 1420.0)
24. Розмір основи на якій розміщено дзеркало:----- (502.0 1210.0)
25. Розмір планки цоколя, що знаходиться під дзеркалом:----- (466.0 50.0)
26. Розмір планки цоколя, що знаходиться під вішаками:----- (664.0 50.0)
27. Розмір планки цоколя, що знаходиться під шафою:----- (564.0 50.0)
28. Розмір передньої внутрішньої стінки шухляди:----- (446.0 90.0)
29. Розмір лівої стінки шухляди:----- (272.0 90.0)
30. Розмір правої стінки шухляди:----- (272.0 90.0)
31. Розмір задньої стінки шухляди:----- (410.0 90.0)
32. Розмір підпорки верхньої частини вішака:-----2шт.----- (150.0 150.0)

Допоміжні розміри передпокою:

1. Розмір ДВП низу шухляди:----- (290.0 446.0)
2. Розмір дзеркала:----- (376.5 756.25)

Рд 1, ствп 1      90%      Windows (CRLF)      ANSI

Рис. 3.4. Текстовий файл із результатами розрахунків заготовок плити передпокою (із врахуванням личківки)

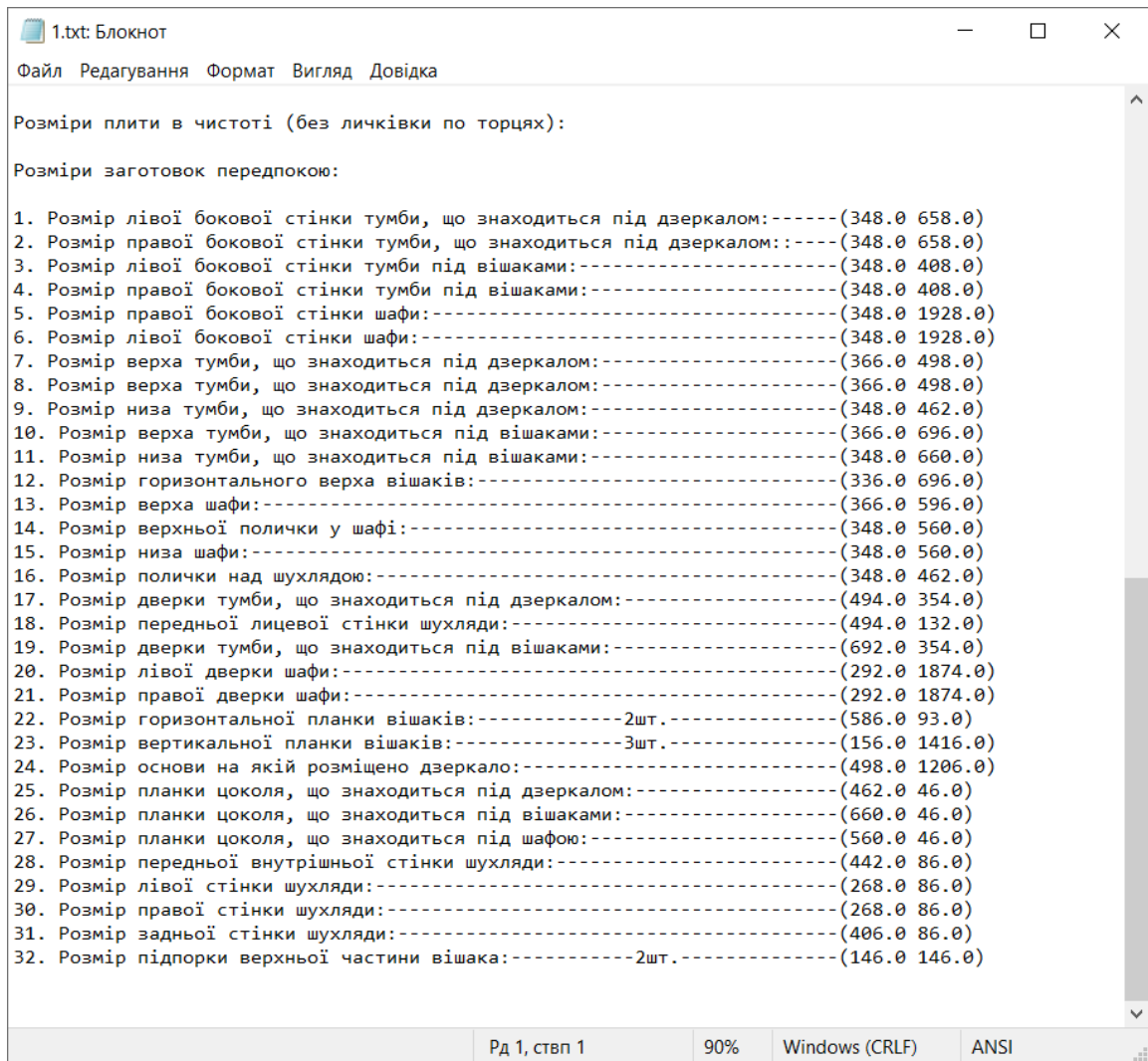


Рис. 3.5. Текстовий файл із результатами розрахунків заготовок плити передпокою (без врахування личківки)

## Програмний код вікна на мові DCL



Наведемо частину програмного коду який дозволяє викликати діалогове вікно написане на мові DCL та отримувати із нього вхідні геометричні данні.

```

;завантаження вікна
(if (<(setq dcl_id (load_dialog "C:/greyd/vikno.dcl")) 0) (exit))
  (if (not (new_dialog "greyd" dcl_id)) (exit))
(action_tile "accept" (strcat "(setq glub_greyd (atof(get_tile
\"glub_greyd\")))"
  "(setq shur_shu (atof(get_tile \"shur_shu\")))"
  "(setq shur_2s (atof(get_tile \"shur_2s\")))"
  "(setq shur_3s (atof(get_tile \"shur_3s\")))"
  "(setq vus_greyd (atof(get_tile \"vus_greyd\")))"
  "(setq ht_z (atof(get_tile \"ht_z\")))"
  "(setq ht_v (atof(get_tile \"ht_v\")))"
  "(setq shur_pl_g (atof(get_tile \"shur_pl_g\")))"
  "(setq shur_pl_v (atof(get_tile \"shur_pl_v\")))"
  "(setq h1 (atof(get_tile \"h1\")))"
  "(setq v_shuch (atof(get_tile \"v_shuch\")))"
  "(setq v_polku (atof(get_tile \"v_polku\")))"
  "(setq zh (atof(get_tile \"zh\")))"
  "(setq h_1_1 (atof(get_tile \"h_1\")))"
  "(setq h1_1_1 (atof(get_tile \"h1_1\")))"
  "(setq b1 (atof(get_tile \"b1\")))"
  "(setq vuvid (get_tile \"b1_1\"))"
  "(done_dialog 0)"))
(start_dialog)
(unload_dialog dcl_id)

```

Програмний код середовища VisualLISP for AutoCAD який виконує побудову 3D каркасу передпокою “Грейд Комфорт” та виводить усі геометричні розрахунки у файл наведений в додатку А.

## ВИСНОВКИ

Розроблено програмне забезпечення, що дозволяє проектувати передпокій.

Для його розробки використовувались мови програмування AutoLISP, VisualLisp та DCL. Ці мови разом з середовищем VisualLISP for AutoCAD дозволяють використовувати функціональні можливості AutoCAD, зокрема його архітектуру. Інтерфейс програми є зрозумілим та зручним, що сприяє швидкій побудові 3Д твердотілої моделі передпокою, яку можна вдосконалити за допомогою функціональних можливостей AutoCAD.

Результатом впровадження даного програмного продукту на підприємстві може бути економія матеріальних та фінансових ресурсів, а також трудозатрат.

Набуті навички під час дипломного проектування дозволяють проектувати різноманітні корпусні меблі та створювати для них додатки у середовищі AutoCAD.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Огляд ринку меблів в Україні: тенденції розвитку [Електронний ресурс] – Режим доступу до ресурсу: <https://rubarbs.com/ua/article/oglyad-rinku-mebliv-v-ukraini-tendentsii-rozvitku> (Дата звернення: 05.06.2023);
2. Дорожня карта конкурентоспроможного розвитку української меблевої промисловості 2018 [Електронний ресурс] – Режим доступу до ресурсу: <https://uafm.com.ua/wp-content/uploads/2018/07/dorozhnya-karta-dlya-rozvytkumeblevogo-sektoru-ukrayiny.pdf> (Дата звернення: 05.06.2023);
3. Програми для 3D-моделювання меблів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.soringpcrepair.com/programs-for-make-furniture-design/> (Дата звернення: 05.06.2023);
4. SolidWorks – світовий стандарт автоматизованого проектування [Електронний ресурс] – Режим доступу до ресурсу: <http://commit.name/index.php?MainShowID=104> (Дата звернення: 05.06.2023);
5. Щербина Ю. Ю., Сахаров О. С., Гондляр О. В., Сівецький В. І. САПР. Програмування на функціональній мові AutoLISP при проектуванні технологічного обладнання., НТУУ «КПІ», 2014 – 120 с.
6. Експертні системи та їх характеристика [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/7144845/page:64/> (Дата звернення: 05.06.2023);
7. Богданова Н. В., Богданов О. В. Математичне моделювання систем і процесів, КПІ ім. Ігоря Сікорського, 2022 – 100 с.
8. Мультикритеріальне програмування [Електронний ресурс] – Режим доступу до ресурсу: [https://ela.kpi.ua/bitstream/123456789/17665/1/Конспект\\_Дослідження%20о перацій.pdf](https://ela.kpi.ua/bitstream/123456789/17665/1/Конспект_Дослідження%20о%20операцій.pdf) (Дата звернення: 05.06.2023);
9. Білогурова Г. В., Самойленко М. І. Конспект лекцій математичне програмування, ХНАМГ, 2009 – 88 с.

10. Montaño La Cruz, F. AutoCAD 2022, Alfaomega Grupo Editor, 2022 – 400 p.
11. Система автоматизованого проектування. Параметричне, поверхневе та твердотільне моделювання [Електронний ресурс] – Режим доступу до ресурсу: <https://rocketmen.com.ua/ua/article/sapr> (Дата звернення: 05.06.2023);
12. Роль України на Європейському деревообробному ринку [Електронний ресурс] – Режим доступу до ресурсу: <https://lira-ukraine.com/ukraine-european-woodworking-market/> (Дата звернення: 05.06.2023);
13. CADArtifex, Willis J., Dogra S. AutoCAD 2022 for Architectural Design: A Power Guide for Beginners and Intermediate Users, CADArtifex Publishing, 2021 – 480 p.
14. Мосіюк О. О. Редактори тривимірної графіки: навчально-методичний посібник, ЖДУ ім. І. Франка, 2022 – 120 с.
15. PRO100 Program Manual [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scribd.com/document/710444911/Pro100-V6-Complete-Manual> (Дата звернення: 05.06.2023).

## ДОДАТКИ

```

(defun c:greyd ( / )

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

  ; Вимикання режимів ORTHO, SNAP, OSNAP
  (setvar "ORTHOMODE" 0)
  (setvar "SNAPMODE" 0)
  (setvar "OSMODE" 0)
  (setvar "3DOSMODE" 0)
  ;Встановлення виглядів, тонування, відключення сітки
  (vl-cmdf "_UCS" "_World")
  (vl-cmdf "_view" "_sei")
  (vl-cmdf "_vscurrent" "_c")
  (vl-cmdf "_grid" "_off")

  ;завантаження вікна
  (if (<(setq dcl_id (load_dialog "C:/grejd/vikno.dcl")) 0) (exit)) ;grejd
  (if (not (new_dialog "grejd" dcl_id)) (exit))
    (action_tile "accept" (strcat "(setq glub_greyd (atof(get_tile
\"glub_greyd\")))"
      "(setq shur_shu (atof(get_tile \"shur_shu\")))"
      "(setq shur_2s (atof(get_tile \"shur_2s\")))"
      "(setq shur_3s (atof(get_tile \"shur_3s\")))"
      "(setq vus_greyd (atof(get_tile \"vus_greyd\")))"
      "(setq ht_z (atof(get_tile \"ht_z\")))"
      "(setq ht_v (atof(get_tile \"ht_v\")))"
      "(setq shur_pl_g (atof(get_tile \"shur_pl_g\")))"
      "(setq shur_pl_v (atof(get_tile \"shur_pl_v\")))"
      "(setq h1 (atof(get_tile \"h1\")))"
      "(setq v_shuch (atof(get_tile \"v_shuch\")))"
      "(setq v_polku (atof(get_tile \"v_polku\")))"
      "(setq zh (atof(get_tile \"zh\")))"
      "(setq h_1_1 (atof(get_tile \"h_1_1\")))"
      "(setq h1_1_1 (atof(get_tile \"h1_1_1\")))"
      "(setq b1 (atof(get_tile \"b1\")))"
      "(setq vuvid (get_tile \"b1_1\"))"
      "(done_dialog 0)"))
  (start_dialog)

  ; ;ТОВЩИНА ПЛИТИ
  (if (= h_1_1 0.0)
    (progn (setq h 18.0))
    (progn (setq h 16.0))
  );if

  ;ТОВЩИНА ОБЛИЦЮВАЛЬНОЇ КРОМКИ
  (if (= h1_1_1 0.0)
    (progn (setq vt7 2.0))
    (progn (setq vt7 1.0))
  );if

  ;;(setq glub_greyd 370.0 vus_greyd 1700.0 shur_shu 402.0 )

  ;=====
  ;;(setq h 18.0);ТОВЩИНА ПЛИТИ*

```

```

;;(setq h1 50.0);висота підняття знизу - цоколь
;;(setq zh 2.0); технологічний зазор (щілини)
  ;;(setq ht_z 680.0);висота тумбочки під зеркалом
  ;;(setq ht_v 430.0);висота тумбочки під вішаками
;=====
(setq h_r 10);ширина для рейки роликової
(setq d_shuch (- glub_greyd 80.0));довжина шуфляди
;;(setq v_shuch 140.0);висота шуфляди
;;(setq v_polku 340.0);висота полки у шафі
;;(setq shur_2s 700.0);ширина секції із вішаками
;;(setq shur_3s 600.0);ширина секції із шафою
;;(setq shur_pl_v 160.0);ширина вертикальної планки вішаків
;;(setq shur_pl_g 97.0);ширина горизонтальної планки вішаків
;;(setq vt7 2.0);товщина облицювальної кромки
(setq rozm 150.0);розмір підпорки верхньої частини вішака
;=====

(setq xyz (getpoint " Вкажіть задній лівий кут Передпокою Грейд Комфорт "))
(setq x (car xyz)
      y (cadr xyz)
      z (nth 2 xyz))

;=====
(defun box (xyz1 x1 y1 z1 / xyz1 x1 y1 z1 )
(vl-cmdf "_ .box" xyz1 "_l" x1 y1 z1 "")
)
;=====
;рисування бокових стінок
;рисування лівої бокової стінки тумбочки під зеркалом
(box xyz (- glub_greyd h) h (- ht_z h)) ;рис 1 бокової стінки
(setq b_sinka1 (list (- glub_greyd h) (- ht_z h)));-----
-----vuvld
;рисування правої бокової стінки тумбочки під зеркалом
(setq t1 (list x (-(+ y shur_shu)h) z ))
(box t1 (- glub_greyd h) h (- ht_z h) );рис 2 бокової стінки
(setq b_sinka2 (list (- glub_greyd h) (- ht_z h)));-----
-----vuvld
;рисування лівої бокової стінки тумбочки під вішаками
(setq t2 (list x (+ y shur_shu) z ))
(box t2 (- glub_greyd h) h (- ht_v h) );рис 3 бокової стінки
(setq b_sinka3 (list (- glub_greyd h) (- ht_v h)));-----
-----vuvld
;рисування правої бокової стінки тумбочки під вішаками
(setq t3 (list x (+(+ y shur_shu)shur_2s) z ))
(box t3 (- glub_greyd h) (* -1 h) (- ht_v h) );рис 4 бокової стінки
(setq b_sinka4 (list (- glub_greyd h) (- ht_v h)));-----
-----vuvld
;рисування правої бокової стінки шафи
(setq t4 (list x (+(+ y shur_shu)shur_2s) z ))
(box t4 (- glub_greyd h) h (- vus_greyd h) );рис 5 бокової стінки
(setq b_sinka5 (list (- glub_greyd h) (- vus_greyd h)));-----
-----vuvld
;рисування лівої бокової стінки шафи
(setq t5 (list x (+(+ (+ y shur_shu)shur_2s)shur_3s) z ))
(box t5 (- glub_greyd h) (* -1 h) (- vus_greyd h) );рис 6 бокової стінки
(setq b_sinka6 (list (- glub_greyd h) (- vus_greyd h)));-----
-----vuvld
;рисування горизонтальних елементів
;верх тумби під зеркалом
(setq t6 (list x y (-(+ z ht_z)h) ) )
(box t6 glub_greyd shur_shu h );рисування

```

```

(setq goruz_elem1 (list glub_greyd shur_shu));-----
--vuvид
;низ тумби під зеркалом
(setq t7 (list x (+ y h) (+ z h1) ))
(box t7 (- glub_greyd h) (apply '- (list shur_shu h h)) h);рисування
(setq goruz_elem2 (list (- glub_greyd h) (apply '- (list shur_shu h h)) ));-----
-----vuvид
;верх тумби під вішаками
(setq t8 (list x (+ y shur_shu) (-(+ z ht_v)h) ))
(box t8 glub_greyd shur_2s h);рисування
(setq goruz_elem3 (list glub_greyd shur_2s));-----
--vuvид
;низ тумби під вішаками
(setq t9 (list x (+(+ y shur_shu)h) (+ z h1) ))
(box t9 (- glub_greyd h) (apply '- (list shur_2s h h)) h);рисування
(setq goruz_elem4 (list (- glub_greyd h) (apply '- (list shur_2s h h)) ));-----
-----vuvид
;верх вішаків
(setq t10 (list (+(+ x h)h) (+ y shur_shu) (-(+ z vus_greyd)h) ))
(box t10 (-(-(- glub_greyd 30.0)h)h) shur_2s h);рисування
(setq goruz_elem5 (list (- glub_greyd 30.0) shur_2s ));-----
-----vuvид
;верх шафи
(setq t11 (list x (+(+ y shur_shu)shur_2s) (-(+ z vus_greyd)h) ))
(box t11 glub_greyd shur_3s h);рисування
(setq goruz_elem6 (list glub_greyd shur_3s));-----
--vuvид
;поличка у шафі
(setq t12 (list x (+(+(+ y shur_shu)shur_2s)h) (-(-(+ z vus_greyd)h)v_polku) ))
(box t12 (- glub_greyd h) (-(- shur_3s h)h) h);рисування
(setq goruz_elem7 (list (- glub_greyd h) (-(- shur_3s h)h)));-----
-----vuvид
;низ шафи
(setq t13 (list x (+(+(+ y shur_shu)shur_2s)h) (+ z h1) ))
(box t13 (- glub_greyd h) (-(- shur_3s h)h) h);рисування
(setq goruz_elem8 (list (- glub_greyd h) (-(- shur_3s h)h)));-----
-----vuvид
;поличка над шухлядою
(setq t14 (list x (+ y h) (-(-(+ z ht_v)v_shuch)h)h) ))
(box t14 (- glub_greyd h) (apply '- (list shur_shu h h)) h);рисування
(setq goruz_elem9 (list (- glub_greyd h) (apply '- (list shur_shu h h)) ));-----
-----vuvид
;рисування вертикальних елементів
;двері тумби що під зеркалом
(setq t15 (list (-(+ x glub_greyd)h) (+ y zh) (+(+ z h1)zh) ))
(box t15 h (- shur_shu (* 2 zh)) (-(-(- ht_v h1)h)(* 2 zh) ));рисування
(setq vertuk_elem1 (list (- shur_shu (* 2 zh)) (-(-(- ht_v h1)h)(* 2 zh) ));-----
-----vuvид
;перід шухляди
(setq t16 (list (-(+ x glub_greyd)h) (+ y zh) (+(+ z ht_v)h)zh) ))
(box t16 h (- shur_shu (* 2 zh)) (- v_shuch (* 2 zh)));рисування
(setq vertuk_elem2 (list (- shur_shu (* 2 zh)) (- v_shuch (* 2 zh) ));-----
-----vuvид
;двері тумби під вішаками
(setq t17 (list (-(+ x glub_greyd)h) (+(+ y shur_shu)zh) (+(+ z h1)zh) ))
(box t17 h (- shur_2s (* 2 zh)) (-(-(- ht_v h1)h)(* 2 zh) ));рисування
(setq vertuk_elem3 (list (- shur_2s (* 2 zh)) (-(-(- ht_v h1)h)(* 2 zh) ));-----
-----vuvид
;ліва дверка шафи
(setq t18 (list (-(+ x glub_greyd)h) (+(+(+ y shur_shu)shur_2s)zh) (+(+ z
h1)zh) ))

```

```

(box t18 h (-(/ shur_3s 2.0)(* 2 zh)) (-(-(- vus_greyd h1)h)(* 2 zh))
);рисування
(setq vertuk_elem4 (list (-(/ shur_3s 2.0)(* 2 zh))(-(-(- vus_greyd h1)h)(* 2
zh)) ));-----vuvuid
;права дверка шафи
(setq t19 (list (-(+ x glub_greyd)h) (+(+(+ y shur_shu)shur_2s)zh)(/ shur_3s
2.0)) (+(+ z h1)zh) ))
(box t19 h (-(/ shur_3s 2.0)(* 2 zh)) (-(-(- vus_greyd h1)h)(* 2 zh))
);рисування
(setq vertuk_elem5 (list (-(/ shur_3s 2.0)(* 2 zh))(-(-(- vus_greyd h1)h)(* 2
zh)) ));-----vuvuid
;1 горизонтальна планка вішаків
(setq t20 (list x (+(+ y shur_shu)(/(- shur_2s (* 3 shur_pl_v))4.0)) (+(+ z
ht_v)100.0) ))
(box t20 h (- shur_2s (* 2 (/(- shur_2s (* 3 shur_pl_v))4.0))) shur_pl_g
);рисування
(setq vertuk_elem6 (list (- shur_2s (* 2 (/(- shur_2s (* 3 shur_pl_v))4.0)))
shur_pl_g ));-----vuvuid
;2 горизонтальна планка вішаків
(setq t21 (list x (+(+ y shur_shu)(/(- shur_2s (* 3 shur_pl_v))4.0)) (-(+ z
vus_greyd)400.0) ))
(box t21 h (- shur_2s (* 2 (/(- shur_2s (* 3 shur_pl_v))4.0))) shur_pl_g
);рисування
(setq vertuk_elem7 (list (- shur_2s (* 2 (/(- shur_2s (* 3 shur_pl_v))4.0)))
shur_pl_g ));-----vuvuid
;1 вертикальна планка вішаків
(setq t22 (list (+ x h) (+(+ y shur_shu)(/(- shur_2s (* 3 shur_pl_v))4.0)) (+(+
z ht_v)100.0) ))
(box t22 h shur_pl_v (-(- vus_greyd ht_v) 100.0) );рисування
(setq vertuk_elem8 (list shur_pl_v (-(- vus_greyd ht_v) 100.0) ));-----
-----vuvuid
;2 вертикальна планка вішаків
(setq t23 (list (+ x h) (+(+ y shur_shu)(* 2(/(- shur_2s (* 3
shur_pl_v))4.0)))shur_pl_v (+(+ z ht_v)100.0) ))
(box t23 h shur_pl_v (-(- vus_greyd ht_v) 100.0) );рисування
(setq vertuk_elem9 (list shur_pl_v (-(- vus_greyd ht_v) 100.0) ));-----
-----vuvuid
;3 вертикальна планка вішаків
(setq t24 (list (+ x h) (+(+ y shur_shu)(* 3(/(- shur_2s (* 3
shur_pl_v))4.0)))(* 2 shur_pl_v) (+(+ z ht_v)100.0) ))
(box t24 h shur_pl_v (-(- vus_greyd ht_v) 100.0) );рисування
(setq vertuk_elem10 (list shur_pl_v (-(- vus_greyd ht_v) 100.0) ));-----
-----vuvuid
;основа під зеркало
(setq t25 (list x y (+(+ z ht_z)60.0) ))
(box t25 h shur_shu (-(- vus_greyd ht_z) 60.0) );рисування
(setq vertuk_elem11 (list shur_shu (-(- vus_greyd ht_z) 60.0) ));-----
-----vuvuid
;зеркало
(setq t26 (list (+ x h) (+ y (/ shur_shu 8.0)) (+(+ z ht_z)60.0)(* /(-(-
vus_greyd ht_z) 60.0)8.0)2.0)) ))
(box t26 5.0 (- shur_shu (* 2(/ shur_shu 8.0))) (-(-(- vus_greyd ht_z) 60.0) (*
3(/(-(- vus_greyd ht_z) 60.0)8.0))) );рисування
(setq zerkalo (list (- shur_shu (* 2(/ shur_shu 8.0)))(-(-(- vus_greyd ht_z)
60.0) (* 3(/(-(- vus_greyd ht_z) 60.0)8.0))) ));-----vuvuid
;цоколь 1 секції
(setq t27 (list (-(-(+ x glub_greyd)h)h) (+ y h) z ))
(box t27 h (-(- shur_shu h)h) h1 );рисування
(setq vertuk_elem12 (list (-(- shur_shu h)h) h1 ));-----
-----vuvuid
;цоколь 2 секції
(setq t28 (list (-(-(+ x glub_greyd)h)h) (+(+ y h)shur_shu) z ))

```

```

(box t28 h (-(- shur_2s h)h) h1 );рисування
(setq vertuk_elem13 (list (-(- shur_2s h)h) h1 ));-----
vuvид
;цоколь 3 секції
(setq t29 (list (-(+ x glub_greyd)h)h) (+(+ y h)shur_shu)shur_2s) z ))
(box t29 h (-(- shur_3s h)h) h1 );рисування
(setq vertuk_elem14 (list (-(- shur_3s h)h) h1 ));-----
vuvид
;рисування боків, заду, низу шуфляд
(setq shur_p_shu (apply '- (list shur_shu (* h 2) h_r h_r)))
(setq t30 (list (-(+ x glub_greyd)h)h) (+(+ y h)h_r) (+(-(+ z ht_v)h)5.0) ))
(box t30 h (apply '- (list shur_shu h h h_r h_r)) (- v_shuch 50.0) );рисування
внутрішнього переду шуфляди
(setq vertuk_elem15 (list (apply '- (list shur_shu h h h_r h_r)) (- v_shuch 50.0)
));-----vuvид
(setq t31 (list (-(+ x glub_greyd)h) (+(+ y h)h_r) (-(+ z ht_v)h) ))
(box t31 (* -1 d_shuch) (apply '- (list shur_shu h h h_r h_r)) 5.0) ;низ шуфляди,
ДВП
(setq nuz_shuxljadu(list d_shuch (apply '- (list shur_shu h h h_r h_r)) ));-----
-----vuvид
(setq t32 (list (-(+ x glub_greyd)h)h) (+(+ y h)h_r) (+(-(+ z ht_v)h)5.0) ))
(box t32 (* -1 (- d_shuch h)) h (- v_shuch 50.0)) ;лів бік шуфляди
(setq liv_bik_shuxljadu(list (- d_shuch h) (- v_shuch 50.0));-----
-----vuvид
(setq t33 (list (-(+ x glub_greyd)h)h) (+(+ y h)h_r) (apply '- (list shur_shu h h
h_r h_r)) (+(-(+ z ht_v)h)5.0)))
(box t33 (* -1 (- d_shuch h)) h (- v_shuch 50.0)) ;лів бік шуфляди
(setq prav_bik_shuxljadu(list (- d_shuch h) (- v_shuch 50.0));-----
-----vuvид
(setq t34 (list (-(+ x glub_greyd)h)d_shuch) (+(+ y h)h_r)h) (+(-(+ z
ht_v)h)5.0) ))
(box t34 h (apply '- (list shur_shu h h h_r h_r h h)) (- v_shuch 50.0) );зад
шуфляди
(setq zad_shuxljadu(list (apply '- (list shur_shu h h h_r h_r h h)) (- v_shuch
50.0));-----vuvид
;розмір підпорки верхньої частини вішака
(setq rozm_p (list rozm rozm));-----vuvид

;створення функцій які формують вивід даних у текстовий файл враховуючи
відсутність кромки
(defun obrizka_kromku1 (len4 / )
  (setq ob_x (- (abs(car len4))vt7)
        ob_y (abs(cadr len4)))
  (setq vuvид_kr1 (list ob_x ob_y))
  )
(defun obrizka_kromku2 (len5 / )
  (setq ob_x (abs(car len5))
        ob_y (- (abs(cadr len5))vt7))
  (setq vuvид_kr2 (list ob_x ob_y))
  )
(defun obrizka_kromku12 (len6 / )
  (setq ob_x (- (abs(car len6))vt7)
        ob_y (- (abs(cadr len6))vt7))
  (setq vuvид_kr12 (list ob_x ob_y))
  )
(defun obrizka_kromku11 (len7 / )
  (setq ob_x (- (- (abs(car len7))vt7)vt7)
        ob_y (abs(cadr len7)))
  (setq vuvид_kr11 (list ob_x ob_y))
  )
(defun obrizka_kromku22 (len8 / )
  (setq ob_x (abs(car len8))

```

```

    ob_y (-(- (abs (cadr len8)) vt7) vt7))
  (setq vuvid_kr22 (list ob_x ob_y))
)
(defun obrizka_kromku1122 (len9 / )
  (setq ob_x (-(- (abs (car len9)) vt7) vt7)
        ob_y (-(- (abs (cadr len9)) vt7) vt7))
  (setq vuvid_kr1122 (list ob_x ob_y))
)
(defun obrizka_kromku112 (len10 / )
  (setq ob_x (-(- (abs (car len10)) vt7) vt7)
        ob_y (- (abs (cadr len10)) vt7))
  (setq vuvid_kr112 (list ob_x ob_y))
)

;запис в тестовий файл результатів розрахунків
(setq b1 1)
(setq vuvid "C:/grejd/1.txt" )
(if (= b1 1)
  (progn

    (setq desk (open (strcat vuvid) "w"))
    (write-line (strcat "Розміри заготовок плити з врахуванням личківки: " )
desk )
    (princ "\n" desk )
    (write-line (strcat "Розміри заготовок передпокою: " ) desk )
    (princ "\n" desk )
    (write-line (strcat "1. Розмір лівої бокової стінки тумби, що знаходиться
під дзеркалом:-----" (vl-prin1-to-string b_sinkal1) ) desk )
    (write-line (strcat "2. Розмір правої бокової стінки тумби, що знаходиться
під дзеркалом:-----" (vl-prin1-to-string b_sinkal2) ) desk )
    (write-line (strcat "3. Розмір лівої бокової стінки тумби під вішаками:----
-----" (vl-prin1-to-string b_sinkal3) ) desk )
    (write-line (strcat "4. Розмір правої бокової стінки тумби під вішаками:---
-----" (vl-prin1-to-string b_sinkal4) ) desk )
    (write-line (strcat "5. Розмір правої бокової стінки шафи:-----
-----" (vl-prin1-to-string b_sinkal5) ) desk )
    (write-line (strcat "6. Розмір лівої бокової стінки шафи:-----
-----" (vl-prin1-to-string b_sinkal6) ) desk )
    (write-line (strcat "7. Розмір верха тумби, що знаходиться під дзеркалом:--
-----" (vl-prin1-to-string goruz_elem1) ) desk )
    (write-line (strcat "8. Розмір верха тумби, що знаходиться під дзеркалом:--
-----" (vl-prin1-to-string goruz_elem1) ) desk )
    (write-line (strcat "9. Розмір низа тумби, що знаходиться під дзеркалом:---
-----" (vl-prin1-to-string goruz_elem2) ) desk )
    (write-line (strcat "10. Розмір верха тумби, що знаходиться під вішаками:--
-----" (vl-prin1-to-string goruz_elem3) ) desk )
    (write-line (strcat "11. Розмір низа тумби, що знаходиться під вішаками:---
-----" (vl-prin1-to-string goruz_elem4) ) desk )
    (write-line (strcat "12. Розмір горизонтального верха вішаків:-----
-----" (vl-prin1-to-string goruz_elem5) ) desk )
    (write-line (strcat "13. Розмір верха шафи:-----
-----" (vl-prin1-to-string goruz_elem6) ) desk )
    (write-line (strcat "14. Розмір верхньої полицки у шафі:-----
-----" (vl-prin1-to-string goruz_elem7) ) desk )
    (write-line (strcat "15. Розмір низа шафи:-----
-----" (vl-prin1-to-string goruz_elem8) ) desk )
    (write-line (strcat "16. Розмір полицки над шухлядою:-----
-----" (vl-prin1-to-string goruz_elem9) ) desk )
    (write-line (strcat "17. Розмір дверки тумби, що знаходиться під
дзеркалом:-----" (vl-prin1-to-string vertuk_elem1) ) desk )
    (write-line (strcat "18. Розмір передньої лицевої стінки шухляди:-----
-----" (vl-prin1-to-string vertuk_elem2) ) desk )

```

```

(write-line (strcat "19. Розмір дверки тумби, що знаходиться під вішаками:--
-----" (vl-prin1-to-string vertuk_elem3) ) desk )
(write-line (strcat "20. Розмір лівої дверки шафи:-----
-----" (vl-prin1-to-string vertuk_elem4) ) desk )
(write-line (strcat "21. Розмір правої дверки шафи:-----
-----" (vl-prin1-to-string vertuk_elem5) ) desk )
(write-line (strcat "22. Розмір горизонтальної планки вішаків:-----
2шт.-----" (vl-prin1-to-string vertuk_elem6) ) desk )
(write-line (strcat "23. Розмір вертикальної планки вішаків:-----
3шт.-----" (vl-prin1-to-string vertuk_elem8) ) desk )
(write-line (strcat "24. Розмір основи на якій розміщено дзеркало:-----
-----" (vl-prin1-to-string vertuk_elem11) ) desk )
(write-line (strcat "25. Розмір планки цоколя, що знаходиться під
дзеркалом:-----" (vl-prin1-to-string vertuk_elem12) ) desk )
(write-line (strcat "26. Розмір планки цоколя, що знаходиться під
вішаками:-----" (vl-prin1-to-string vertuk_elem13) ) desk )
(write-line (strcat "27. Розмір планки цоколя, що знаходиться під шафою:---
-----" (vl-prin1-to-string vertuk_elem14) ) desk )
(write-line (strcat "28. Розмір передньої внутрішньої стінки шухляди:-----
-----" (vl-prin1-to-string vertuk_elem15) ) desk )
(write-line (strcat "29. Розмір лівої стінки шухляди:-----
-----" (vl-prin1-to-string liv_bik_shuxljadu) ) desk )
(write-line (strcat "30. Розмір правої стінки шухляди:-----
-----" (vl-prin1-to-string prav_bik_shuxljadu) ) desk )
(write-line (strcat "31. Розмір задньої стінки шухляди:-----
-----" (vl-prin1-to-string zad_shuxljadu) ) desk )
(write-line (strcat "32. Розмір підпорки верхньої частини вішака:-----
-2шт.-----" (vl-prin1-to-string rozm_p) ) desk )
(princ "\n" desk )
(write-line (strcat "Допоміжні розміри передпокою: " ) desk )
(princ "\n" desk )
(write-line (strcat "1. Розмір ДВП низу шухляди:-----
-----" (vl-prin1-to-string nuz_shuxljadu) ) desk )
(write-line (strcat "2. Розмір дзеркала:-----
-----" (vl-prin1-to-string zerkalo) ) desk )
(princ "\n" desk )
(princ "\n" desk )
(princ "\n" desk )
(write-line (strcat "Розміри плити в чистоті (без личківки по торцях): " )
desk )
(princ "\n" desk )
(write-line (strcat "Розміри заготовок передпокою: " ) desk )
(princ "\n" desk )
(write-line (strcat "1. Розмір лівої бокової стінки тумби, що знаходиться
під дзеркалом:-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka1)) ) desk )
(write-line (strcat "2. Розмір правої бокової стінки тумби, що знаходиться
під дзеркалом:-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka2)) ) desk )
(write-line (strcat "3. Розмір лівої бокової стінки тумби під вішаками:----
-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka3)) ) desk )
(write-line (strcat "4. Розмір правої бокової стінки тумби під вішаками:---
-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka4)) ) desk )
(write-line (strcat "5. Розмір правої бокової стінки шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka5)) ) desk )
(write-line (strcat "6. Розмір лівої бокової стінки шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 b_sinka6)) ) desk )
(write-line (strcat "7. Розмір верха тумби, що знаходиться під дзеркалом:--
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem1)) ) desk
)
(write-line (strcat "8. Розмір верха тумби, що знаходиться під дзеркалом:--
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem1)) ) desk
)

```

```

(write-line (strcat "9. Розмір низа тумби, що знаходиться під дзеркалом:---
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem2)) ) desk
)
(write-line (strcat "10. Розмір верха тумби, що знаходиться під вішаками:--
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem3)) ) desk
)
(write-line (strcat "11. Розмір низа тумби, що знаходиться під вішаками:---
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem4)) ) desk
)
(write-line (strcat "12. Розмір горизонтального верха вішаків:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem5)) ) desk
)
(write-line (strcat "13. Розмір верха шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem6)) ) desk
)
(write-line (strcat "14. Розмір верхньої полицки у шафі:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem7)) ) desk
)
(write-line (strcat "15. Розмір низа шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem8)) ) desk
)
(write-line (strcat "16. Розмір полицки над шухлядою:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 goruz_elem9)) ) desk
)
(write-line (strcat "17. Розмір дверки тумби, що знаходиться під
дзеркалом:-----" (vl-prin1-to-string (obrizka_kromkul122
vertuk_elem1)) ) desk )
(write-line (strcat "18. Розмір передньої лицевої стінки шухляди:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem2)) )
desk )
(write-line (strcat "19. Розмір дверки тумби, що знаходиться під вішаками:--
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem3)) )
desk )
(write-line (strcat "20. Розмір лівої дверки шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem4)) )
desk )
(write-line (strcat "21. Розмір правої дверки шафи:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem5)) )
desk )
(write-line (strcat "22. Розмір горизонтальної планки вішаків:-----
2шт.-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem6)) )
desk )
(write-line (strcat "23. Розмір вертикальної планки вішаків:-----
3шт.-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem8)) )
desk )
(write-line (strcat "24. Розмір основи на якій розміщено дзеркало:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem11)) )
desk )
(write-line (strcat "25. Розмір планки цоколя, що знаходиться під
дзеркалом:-----" (vl-prin1-to-string (obrizka_kromkul122
vertuk_elem12)) ) desk )
(write-line (strcat "26. Розмір планки цоколя, що знаходиться під
вішаками:-----" (vl-prin1-to-string (obrizka_kromkul122
vertuk_elem13)) ) desk )
(write-line (strcat "27. Розмір планки цоколя, що знаходиться під шафою:---
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem14)) )
desk )
(write-line (strcat "28. Розмір передньої внутрішньої стінки шухляди:-----
-----" (vl-prin1-to-string (obrizka_kromkul122 vertuk_elem15)) )
desk )

```

```

        (write-line (strcat "29. Розмір лівої стінки шухляди:-----
-----" (vl-prin1-to-string (obrizka_kromku1122 liv_bik_shuxljadu))
) desk )
        (write-line (strcat "30. Розмір правої стінки шухляди:-----
-----" (vl-prin1-to-string (obrizka_kromku1122 prav_bik_shuxljadu))
) desk )
        (write-line (strcat "31. Розмір задньої стінки шухляди:-----
-----" (vl-prin1-to-string (obrizka_kromku1122 zad_shuxljadu)) )
desk )
        (write-line (strcat "32. Розмір підпорки верхньої частини вішака:-----
-2шт.-----" (vl-prin1-to-string (obrizka_kromku1122 rozm_p)) ) desk )
        (princ "\n" desk )
        (close desk)
        )
);if
; Відновлення стану режимів ORTHO, SNAP , OSNAP
(setvar "ORTHOMODE" p_ortho)
(setvar "SNAPMODE" p_snapm)
(setvar "OSMODE" p_osmode)
(setvar "3DOSMODE" p_3dosmode)

(princ)
);defun

```