

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

з курсового проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему “ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРВІСНИХ БІБЛІОТЕК
AUTOCAD”

Виконав(ла): студент(ка) 3 курсу, групи КНТ-221

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

КІРІЄНКО А. С.

(ПРІЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А. В.

(ПРІЗВИЩЕ та ініціали)

Рецензент АНДРЕЄВ М. О.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА КУРСОВИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КІРІЄНКА Андрія Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) “Програмна реалізація сервісних бібліотек AutoCAD”,
“Гвинт невиспадаючий з напівпотайною головою” ДСТУ 10340-80

керівник проєкту (роботи) доцент, ПАРХОМЕНКО Анжеліка Володимирівна,

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

затверджені наказом закладу вищої освіти від “04” березня 2024 року №

2. Строк подання студентом проєкту (роботи) 20 травня 2024 року

3. Вихідні дані до проєкту (роботи) “Гвинт невиспадаючий з напівпотайною
головою” ДСТУ 10340-80

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз сучасного стану та перспектив розвитку CAD/CAM – Систем. 2. Розширення функціональних можливостей AUTOCAD засобами AUTOLISP. 3. Програмна реалізація спеціалізованих сервісних бібліотек в середовищі AUTOCAD.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	ЛИПОВЕЦЬ М.В., асистент		

7. Дата видачі завдання «04» березня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 3
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Андрій КІРІЄНКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Анжеліка ПАРХОМЕНКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до курсової роботи бакалавра: 65 с., 4 табл., 21 рис., 1 дод., 11 джерел.

САПР, БІБЛІОТЕКА, ДСТУ, ПРОЄКТУВАННЯ, КРЕСЛЕННЯ, AUTOCAD, LISP, ІНТЕРФЕЙС, МОДЕЛЮВАННЯ, РОЗРОБКА, ГВИНТ, СЕРЕДОВИЩЕ.

Мета роботи – програмна реалізація сервісних бібліотек в середовищі AutoCAD.

Об’єкт розробки – гвинт неவிпадающий з напівпотайною головою” ДСТУ 10340-80.

Матеріали, методи та технічні засоби: мова програмування Lisp, персональний комп’ютер з процесором Intel Core 5 10th GEN під управлінням операційної системи Microsoft Windows 11.

Результати. Створено кресленик гвинта за розмірами вихідних даних.

Висновки. Розроблено програмну реалізацію сервісних бібліотек AutoCAD неவிпадающего гвинта з напівпотайною головою ДСТУ 10340-80.

Галузь використання – використання для кріплення деталей у різних видах машин та механізмів, де потрібна висока надійність з’єднань, в приладах та електронному обладнанні для забезпечення точного і надійного кріплення компонентів, в конструкціях літальних апаратів, де важливо уникнути випадання кріпильних елементів при експлуатації, для кріплення елементів конструкцій, де необхідне надійне і приховане кріплення, в різних видах військової техніки, де важлива висока надійність та безпека кріпильних з’єднань.

ЗМІСТ

Перелік скорочень та умовних позначок	6
Вступ.....	7
1 Аналіз сучасного стану та перспектив розвитку CAD/CAM – Систем.....	8
1.1 Огляд сучасних САПР, що використовуються у машинобудівній галузі .	8
1.2 Структура, призначення та функціональні можливості САПР AutoCAD .	9
2 Розширення функціональних можливостей AUTOCAD засобами AUTOLISP.....	17
2.1 Особливості мови LISP.....	17
2.1.1 Формальний опис мови	17
2.1.2 Базові відомості.....	17
2.2 Мова AutoLISP як засіб адаптації AutoCAD до різноманітних застосувань.....	18
3 Програмна реалізація спеціалізованих сервісних бібліотек в середовищі AUTOCAD	21
3.1 Аналіз вхідної інформації до роботи	21
3.2 Схема алгоритму методики створення сервісної бібліотеки стандартних елементів	23
3.3 Розробка структури сервісного застосунку.....	24
3.4 Опис особливостей програмної реалізації бібліотеки.....	25
3.5 Порадник для користувача.....	27
3.6 Порадник для програміста	34
3.6.1 Підключення директорію з програмним забезпеченням до конфігураційних налаштувань середовища розробки	34
3.6.2 Створення та підключення вкладки до робочого простору	36
3.6.3 Порадник для підтримки програмного застосунку	41
Висновки	42
Перелік джерел посилань	43
Додаток А Текст програми.....	45

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ADEM	– Advanced DEM;
AMD	– Advanced Micro Devices;
BIM	– Building Information Modeling;
CAD	– Computer-Aided Design;
CADD5	– Computer-Aided Design and Drafting System;
CAE	– Computer-Aided engineering;
CAM	– Computer-Aided Manufacturing;
CATIA	– Computer-Aided Three-dimensional Interactive Application;
DWG	– Drawing;
EOM	– Engineer-to-Order Manufacturing;
VPN	– Virtual Private Network;
ДСТУ	– державний стандарт України;
ПК	– персональний комп'ютер;
САПР	– система автоматизованого проектування;
ЧПК	– числове програмне керування.

ВСТУП

У сучасному світі інженерно-технічної творчості та розробки продукції актуальність використання комп'ютерних програм із сервісними бібліотеками для автоматизації процесів проектування та моделювання важко переоцінити. Зокрема, у контексті проектування невинпадаючих гвинтів з напівпотайною головою, вирішення відповідних завдань вимагає високої точності, ефективності та використання стандартів якості. Державний стандарт України 10340-80 встановлює вимоги до таких з'єднань та фіксуючих елементів, в тому числі невинпадаючих гвинтів, що надає додаткову вагу та значимість розробці програмної реалізації для автоматизації цього процесу.

Програмна реалізація сервісних бібліотек AutoCAD для моделювання невинпадаючих гвинтів з напівпотайною головою з використанням ДСТУ 10340-80 є актуальним завданням, що вирішує важливі завдання автоматизації та оптимізації процесів проектування в інженерно-технічній сфері. У даній пояснювальній записці буде розглянуто процес розробки та особливості використання цієї програмної реалізації, що відповідає сучасним вимогам та стандартам у галузі інженерного проектування.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ПЕРСПЕКТИВ РОЗВИТКУ CAD/CAM – СИСТЕМ

1.1 Огляд сучасних САПР, що використовуються у машинобудівній галузі

У сучасному виробництві широке поширення одержали системи автоматизованого проєктування (САПР), які дозволяють проєктувати технологічні процеси з меншими витратами часу та засобів, зі збільшенням точності спроектованих процесів і програм обробки, що скорочує витрати матеріалів та час обробки, завдяки тому, що режими обробки також розраховуються та оптимізуються за допомогою ЕОМ [1].

Технічне забезпечення САПР засновано на використанні обчислювальних мереж і телекомунікаційних технологій, персональних комп'ютерів та робочих станцій.

Математичне забезпечення САПР характеризується різноманітністю методів обчислювальної математики, статистики, математичного програмування, дискретної математики, штучного інтелекту. Програмні комплекси САПР відносяться до числа найбільш складних сучасних програмних систем, заснованих на операційних системах Unix, Windows, мовах програмування C, C++, Java і інших, сучасних CASE технологіях, реляційних і об'єктно-орієнтованих системах керування базами даних, стандартах відкритих систем і обміну даними в комп'ютерних середовищах.

Проєктування, при якому всі проєктні рішення або їхня частина одержують шляхом взаємодії людини та ЕОМ, називають автоматизованими на відміну від ручного (без використання ЕОМ) або автоматичного (без участі людини на проміжних етапах). Система, що реалізує автоматизоване проєктування, являє собою систему автоматизованого проєктування (в англomовному написанні CAD System). САПР (або CAD) звичайно використовуються разом із системами автоматизації інженерних розрахунків і аналізу САЕ. Дані із CAD- систем передаються в CAM – систему автоматизованої розробки програм обробки деталей для верстатів.

CAE – автоматизоване конструювання, використання спеціального програмного забезпечення для проведення інженерного аналізу міцності та інших технічних характеристик компонентів, виконаних у системах автоматизованого проєктування. Програми автоматизованого конструювання дозволяють здійснювати динамічне моделювання, перевірку та оптимізацію виробів і засобів їхнього виробництва.

CAM – автоматизоване виробництво. Термін використовується для позначення програмного забезпечення, основною метою якого є створення програм для керування верстатами зі ЧПК. Вхідними даними CAM- системи є геометрична модель виробу, розроблена в системі автоматизованого проєктування. У процесі інтерактивної роботи із тривимірною моделлю в CAM системі інженер визначає траєкторії руху різального інструменту по заготівлі виробу, які потім автоматично верифікуються, візуалізуються (для візуальної перевірки коректності) і обробляються постпроцесором для одержання програми керування конкретним верстатом.

1.2 Структура, призначення та функціональні можливості САПР AutoCAD

САПР – система, що поєднує технічні засоби, математичне й програмне забезпечення, параметри й характеристики яких вибирають із максимальним обліком особливостей завдань інженерного проєктування й конструювання [2]-[4].

Структурними складовими САПР є підсистеми, що володіють всіма властивостями систем і створені як самостійні системи. Це виділені по деяких ознаках частини САПР, що забезпечують виконання деяких закінчених проєктних завдань із одержанням відповідних проєктних рішень і проєктних документів.

По призначенню підсистеми САПР розділяють на два види: ті, що проєктують і ті, що обслуговують.

До систем, що проєктують, відносять підсистеми, що виконують проєктні процедури й операції, наприклад:

- підсистема компоновання виробу;
- підсистема проєктування складальних одиниць;
- підсистема проєктування деталей;
- підсистема проєктування схеми керування;
- підсистема технологічного проєктування.

До систем, що обслуговують, відносять підсистеми, призначені для підтримки працездатності підсистем, що проєктують, наприклад:

- підсистема графічного відображення об'єктів проєктування;
- підсистема документування;
- підсистема інформаційного пошуку й ін.

Залежно від відношення до об'єкта проєктування розрізняють два види підсистем, що проєктують:

- об'єктно-орієнтовні (об'єктні);
- об'єктно-незалежні (інваріантні).

До об'єктних підсистем відносять підсистеми, що виконують одну або кілька проєктних процедур або операцій, безпосередньо залежних від конкретного об'єкта проєктування, наприклад:

- підсистема проєктування технологічних систем;
- підсистема моделювання динаміки, проєктованої конструкції та ін.

До інваріантних підсистем відносять підсистеми, що виконують уніфіковані проєктні процедури й операції, наприклад:

- підсистема розрахунків деталей виробу;
- підсистема розрахунків режимів різання;
- підсистема розрахунку техніко-економічних показників й ін.

Процес проєктування реалізується в підсистемах у вигляді певної послідовності проєктних процедур й операцій. проєктна процедура відповідає частині проєктної підсистеми, у результаті виконання якої приймається деяке проєктне рішення. Вона складається з елементарних проєктних операцій, має твердо встановлений порядок їхнього виконання й спрямована на досягнення

локальної мети в процесі проектування. Під проєктною операцією розуміють умовно виділену частину проєктної процедури або елементарну дію, чинену конструктором у процесі проектування. Прикладами проєктних процедур можуть служити процедури розробки схеми компоновки верстата, технології обробки виробів і т.п., а прикладами проєктних операцій – розрахунок припусків, рішення якого-небудь рівняння й інші.

Структура процесу проектування [2],[3],[5] має такі ієрархічні рівні:

- макрорівень;
- макрорівень;
- функціонально-логічне проектування;
- системне проектування.

Залежно від функціональних можливостей, набору модулів і структурної організації CAD/CAE/CAM – системи можна умовно розділити на три класи [2]-[4]: легкі, середні і важкі системи.

Легкі системи – це перший в історичному розвитку, що склався, клас систем. До цієї категорії можна віднести такі системи, як AutoCAD, CAD-KEY, Personal Designer, ADEM, КОМПАС. Вони, як правило, використовуються на персональних комп'ютерах окремими користувачами. Такі системи призначені в основному для якісного виконання креслень. Також вони можуть використовуватися для двомірного моделювання і нескладних тривимірних побудов. Ці системи досягли останнім часом високого рівня досконалості. Вони прості у використуванні, містять безліч бібліотек стандартних елементів, підтримують різні стандарти оформлення графічної документації.

Системи середнього класу з'явилися порівняно недавно і являють собою недорогі тривимірні CAD системи. До них відносяться системи AMD, Solid Edge, Solid Works і т.д. Їх поява пов'язана із збільшенням потужності персональних комп'ютерів і розвитком операційної системи. З їх допомогою можна вирішувати до 80% типових машинобудівних задач, не привертаючи могутні і дорогі CAD/CAM системи важкого класу. Більшість систем середнього класу ґрунтується на тривимірному твердотільному моделюванні. Вони дозволяють проєктувати більшість деталей загального машинобудування, складальні

одиниці середнього рівня складності, виконувати спільну роботу групам конструкторів. В цих системах можливо проводити аналіз перетинів і зазорів в складках.

Системи важкого класу надають повний набір інтегрованих засобів проєктування, виробництва, аналізу виробів. В цю категорію систем потрапляють, наприклад, CATIA, Unigraphics, Pro/ENGINEER, CADD5, EUCLID, Cimatron. Вони використовують могутні апаратні засоби, як правило, робочі станції з операційною системою UNIX. Системи важкого класу дозволяють вирішувати широкий спектр конструкторсько-технологічних задач.

Окрім функцій, доступних системам середнього класу, важким CAD/CAM системам доступні:

- проєктування деталей найскладнішого типу, що містять дуже складні поверхні;
- виконання побудови поверхонь за наслідками обміру реальної деталі, виконання згладжування поверхонь і складних сполучень;
- проєктування масивних складок, що вимагають ретельної компоновки і містять елементи інфраструктури (кабельні джгути, трубопроводи);
- робота з складними складками в режимі варіантного аналізу для швидкого перегляду і оцінки якості компоновки виробу.

До функціональностей можливостей САПР можна віднести 8 пунктів [6]

Пункт 1. Доступ до спеціалізованих наборів інструментів. Багато хто вважає AutoCAD просто інструментом для створення 2D і 3D-креслень. Незважаючи на те, що він відмінно справляється з цими функціями, сучасний AutoCAD пропонує набагато більше, особливо з додаванням наборів інструментів. Тільки AutoCAD пропонує сім галузевих наборів інструментів з перевагами для користувачів у більшості галузей.

Ці набори інструментів включають:

- набір інструментів для архітектури;
- набір механічних інструментів;
- набір інструментів 3D мапи;

- набір інструментів MEP;
- набір електричних інструментів;
- набір інструментів Plant 3D;
- набір інструментів для растрового дизайну.

Одне нове дослідження продуктивності, розроблене Autodesk та доручене незалежному консультанту, показало, що одні й ті самі завдання проєктування виконували в середньому на 95% швидше за допомогою набору інструментів для роботи з електрикою (залежно від рівня знань користувача при роботі з програмним забезпеченням набору інструментів для роботи з електрикою, а також на основі досвіду та навчання).

Набір інструментів Raster Design я вважаю вкрай важливим для всіх, кому необхідно включити відскановані деталі або вбудовані елементи до наборів планів.

За допомогою Raster Design можна виправити багато найбільш поширених проблем, що виникають під час роботи з растровими зображеннями. Це може бути будь-що: від сканів зі спотвореною віссю XY до зображень, які необхідно обрізати або видалити плями. Все це можна зробити легко, не виходячи з AutoCAD (або обраного вами галузевого набору інструменту). За допомогою Raster Design ви можете перетворити растрові лінії на лінії AutoCAD або лінії AutoCAD на растрові лінії всього кількома клацаннями миші.

Другий набір інструментів це Map 3D. Хоча набір інструментів Map 3D містить безліч геоінформаційних (ГІС) функцій, він пропонує набагато більше. Безліч можливостей відкриваються при підключенні креслення AutoCAD до цієї бази даних. Можливо, найбільш примітним є використання Map 3D як легкого інструменту автоматизованого керування об'єктами (CAFM) для відстеження та керування простором усередині будівель.

Пункт 2. Новий досвід встановлення та розгортання.

Незважаючи на все, чим славиться програмне забезпечення Autodesk, простий процес інсталяції не входить до їх числа. Так було до випуску AutoCAD 2022 та решти портфеля програмного забезпечення Autodesk 2022 року для

настільних ПК. Сьогоднішній досвід установки краще, ніж будь-коли, і навіть включає деякі неймовірні функції, орієнтовані на менеджера САПР.

За допомогою функції «Вибіркова установка» ви можете вибрати набори інструментів AutoCAD та/або інше програмне забезпечення Autodesk, яке хочете встановити, і створити розгортання користувача тільки з цим програмним забезпеченням. Більше того, можна включити свій профіль AutoCAD (ARG) у вибірккову установку, щоб AutoCAD автоматично налаштовувався з урахуванням специфіки вашої компанії.

Об'єднати це з Microsoft Endpoint Configuration Manager (раніше System Center Configuration Manager і Systems Management Server), і можна поширити AutoCAD на сотні робочих станцій, не торкаючись безпосередньо жодної з них.

Пункт 3. Просте ліцензування програмного забезпечення – сервер ліцензій не потрібний. Завдяки пандемії все більше людей працюють віддалено. Традиційно для віддаленого використання таких інструментів, як AutoCAD, у корпоративному середовищі потрібно не забути перевірити ліцензію перед виходом з офісу або отримати доступ до VPN, щоб ви могли перевірити її.

Завдяки переходу на підписку і модель іменованого користувача менеджером САПР та кінцевим користувачам більше не потрібно боротися з серверами ліцензій. Оскільки підписки на такі інструменти, як AutoCAD, призначаються конкретній людині, куди б ця людина не пішла, те саме можна сказати і про її AutoCAD. Сервер ліцензій не потрібний.

Пункт 4. Керування стандартами САПР і застосування їх. Серед менеджерів САПР є приказка, що стандарти САПР схожі на зубні щітки. Вони є у всіх, але ніхто не хоче використовувати чужу. Якщо це схоже на вашу фірму та її використання AutoCAD, ви можете перевірити Диспетчер стандартів САПР, вбудований у програмне забезпечення.

Диспетчер стандартів САПР дозволяє автоматично перевіряти креслення наявність порушень стандартів. Від стилів тексту до типів ліній, від шарів до мультивиносок і розмірів Менеджер стандартів САПР може попередити вашу команду, коли вони відхиляються від стандартів вашої компанії.

Диспетчер стандартів САПР – потужна частина програмного забезпечення, яка часто не береться до уваги. Навіть для тих, хто знає про його існування, мало хто знає всі неймовірні трюки, які в нього є в рукаві. Серед них такі функції, як засіб перевірки стандартів пакетної обробки.

Пункт 5. My Insights у AutoCAD 2022.1+. Використовуючи консультативний механізм, вбудований у програмне забезпечення, він аналізує, як людина використовує програмне забезпечення, а потім надає персоналізовані ідеї, що визначають можливості підвищення кваліфікації в AutoCAD.

Пункт 6. Співпраця. Поліпшення спільної роботи є постійним пріоритетом для проєктних груп, але ще ніколи ця потреба не була більш важливою, ніж сьогодні в нашому постпандемічному світі, що розвивається. Вже небезпечно припускати, що вся команда проєкту перебуватиме в одному місці протягом усього процесу проєктування. Крім того, мало гарантій, що вся проєктна група працюватиме над проєктом одночасно.

У результаті сьогоднішні потреби у співпраці не тільки інші, а й більш вимогливі, ніж будь-коли раніше. Нам потрібно більше, ніж просто можливість додати коментар або відправити малюнок колезі.

Як і слід було очікувати, сучасний AutoCAD легко інтегрується з хмарними інструментами Autodesk для спільної роботи, такими як Autodesk Docs та BIM 360. Хоча ці інструменти безцінні для людей, які використовують ці інструменти, як щодо багатьох інших хмарних рішень для зберігання даних?

Що ж, сучасний AutoCAD інтегрується і з ними. Від Dropbox та OneDrive до Box та Google Drive – AutoCAD забезпечує гнучкість роботи з хмарним сховищем, яке найкраще підходить для вас та вашої організації.

Звичайно, незважаючи на те, що місце, де зберігаються файли DWG, важливо, як щодо всіх інших спільних завдань, що відбуваються у проєктах? Ось де такі функції, як Shared Views і Trace, можуть виявитися безцінними для вас, команди та клієнтів.

Пункт 7. Сумісність. Від Revit до Inventor і в багатьох проміжних областях у багатьох сучасних проєктах використовується AutoCAD і один або

кілька додаткових інструментів. Саме тут оперативна сумісність має вирішальне значення.

Кожен файл DWG, створений за допомогою AutoCAD, є TrustedDWG. Це ключова функція AutoCAD і AutoCAD LT, оскільки вона перевіряє сумісність та цілісність файлу DWG при відкритті. Це допомагає забезпечити можливість відкриття та використання створених вами файлів DWG людьми, яким ви їх відправляєте.

Окрім TrustedDWG, AutoCAD також включає безліч вхідних даних для взаємодії. Дуже важливою є можливість посилаючись на координаційні моделі BIM всередині файлу DWG. Використання цієї функції дозволить відображати дані з різних джерел, включаючи Revit.

Пункт 8. Спільнота Autodesk. Останнє, але не менш важливе, це спільнота Autodesk – це набір ресурсів, за допомогою яких користувачі програмного забезпечення Autodesk можуть навчатися, ставити питання та спілкуватися з іншими експертами в галузі.

Деякі ресурси спільноти, якими я найчастіше користуюся – це форуми Autodesk, Autodesk Group Network та Autodesk University.

2 РОЗШИРЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ AUTOCAD ЗАСОБАМИ AUTOLISP

2.1 Особливості мови LISP

Lisp – мова програмування загального призначення з підтримкою парадигм функціонального та процедурного програмування. Вихідна інформація записується у вигляді списків [7].

Мову програмування Lisp було розроблено в кінці 1950-х у Массачусетському Технологічному Інституті для дослідження проблем штучного інтелекту. Але, через потужність закладених принципів, мова програмування Lisp також придатна для багатьох інших застосувань.

2.1.1 Формальний опис мови

Комп'ютерна програма на Lisp представляє рекурсивну функцію символічних виразів, яка будується аналогічно арифметичним функціям із елементарних з допомогою умовного оператора та операції суперпозиції. Умовний оператор має вигляд $(p_1 \rightarrow l_1; \dots, p_n \rightarrow l_n)$. Результатом його виконання буде вираз l_i , якщо p_i є істинним.

Існує п'ять елементарних функцій:

- atom: булева функція, яка визначає чи є досліджуваний вираз атомом – неподільною одиницею інформації;
- car, cdr: функції, які виокремлюють перший елемент та хвіст (список із всіх елементів окрім першого) відповідно;

Окрім елементарних функцій визначаються ряд складніших функцій, які будуються на їхній основі.

2.1.2 Базові відомості

LISP означає LIST Processing (обробка списків), мова програмування працює із списками (та списками списків) розміщуючи їх між дужками. Дужки

визначають межі списку. Списки є базисом мови програмування Лісп. Мова програмування Лісп була однією із перших мов програмування з автоматичним прибиранням сміття із пам'яті.

Однією з переваг Lisp є те, що кожна змінна може виступати як рядок символів (власне ім'я), посилання на значення, структура даних або функція. Саме останній факт зробив цю мову дуже зручною при розробці лінгвістичних програм, особливо для природних мов з чіткою структурою речення (наприклад, англійська). В таких мовах кожне слово, його зміст/сенс/імператив, можна інтерпретувати як функцію від слів, що знаходяться на чітко визначених позиціях у реченні, до того ж ці позиції визначаються самим цим словом. Приклад системи, що побудована на цій ідеї можна знайти в книзі Т. Вінограда «Програма яка розуміє природну мову». Ця система реалізує діалог з користувачем природною мовою. Користувач бачить перед собою стіл з деякими предметами різного кольору, і може віддавати накази природною мовою, про перенесення якогось предмету. При цьому система (маніпулятор) сама визначає що треба зняти з цього предмету, який предмет на яких можна класти (на піраміду вже нічого не покладеш), і якщо є неоднозначність у виборі предмету ставить уточнювальні питання. Також реалізовано контекстне посилання займенників по тексту діалогу (користувач може сказати: «перестав той куб туди-то», і система з тексту діалогу може визначити, який саме «той куб»). Це імперативи. Також система може відповідати на питання. Вивід відповіді схожий до прологівського. Також є можливість користувачу висловлювати декларативи – вносити нові знання про об'єкти, наприклад, надавати їм імена. Хоча словник і база знань відносно не велика, але вона має можливості до розширення.

2.2 Мова AutoLISP як засіб адаптації AutoCAD до різноманітних застосувань

AutoLISP є конкретною реалізацією мови програмування LISP, вкладеної у систему AutoCAD. AutoLISP пропонує користувачам і розробникам системи

AutoCAD можливість написання макровизначень за допомогою орієнтованих на машинну графіку мовних засобів високого рівня [8] .

Мова AutoLISP розрізняє декілька типів даних: списки, символи, рядки, дійсні числа, цілі числа, дескриптори файлів, назви елементів AutoCAD, назви елементів меню, підпрограми (вкладені функції).

Цілі числа займають 16 біт пам'яті і лежать у діапазоні від мінус 32 768 до плюс 32 767. Дійсні числа зберігаються у форматі плаваючих чисел з подвійною точністю, рядки можуть мати довільну довжину, оскільки пам'ять для них виділяється динамічно.

AutoLISP має набір вкладених функцій, призначених для програмування двох і тривимірних графічних об'єктів. Для роботи з координатами точок використовують такі позначення:

- точка двовимірного простору зображується у вигляді списку, що складається з пари чисел (X, Y), наприклад (3.40000 7.52000), перше число є X-координатою, а друге – Y-координатою;
- точка тривимірного простору зображується у вигляді списку, що складається з трьох чисел.

AutoLISP має набір заздалегідь визначених функцій, для виклику яких потрібно ввести ім'я й аргументи (якщо вони потрібні).

Функція переводить AutoCAD у режим очікування введення користувача. При цьому в нижньому рядку екрана виводиться підказка.

Мова програмування AutoLISP – це мова, яка використовується для автоматизації та розширення функціональності програмного забезпечення AutoCAD. AutoCAD – це один з найпопулярніших пакетів CAD, і він часто використовується в інженерних, архітектурних та будівельних галузях.

AutoLISP дозволяє користувачам створювати спеціальні програми (або "скрипти"), які можуть автоматизувати рутинні завдання, створювати нові функції та розширювати можливості AutoCAD. Це забезпечує більш ефективне та продуктивне використання програмного забезпечення, а також сприяє адаптації AutoCAD до різних вимог та застосувань.

Основні можливості та переваги використання AutoLISP у AutoCAD включають:

- Автоматизація завдань: AutoLISP дозволяє створювати скрипти для автоматизації рутинних завдань, таких як створення складних геометричних форм, розмітка архітектурних елементів, генерація звітів та інше.
- Розширення функціональності: Можна створювати нові функції та команди, які доповнюють можливості AutoCAD та відповідають конкретним потребам користувача.
- Підтримка користувачів: AutoLISP дозволяє створювати власні користувацькі інтерфейси та панелі інструментів, що спрощує роботу з програмним забезпеченням та забезпечує його більш зручне використання.
- Розв'язання специфічних завдань: AutoLISP може використовуватися для створення спеціалізованих рішень для конкретних галузей, таких як архітектура, інженерія, геодезія, механіка та інше.

Загалом, AutoLISP є потужним інструментом для адаптації та розширення функціональності AutoCAD, що дозволяє вирішувати різноманітні завдання та вимоги користувачів у сфері проектування та конструювання.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СПЕЦІАЛІЗОВАНИХ СЕРВІСНИХ БІБЛІОТЕК В СЕРЕДОВИЩІ AUTOCAD

3.1 Аналіз вхідної інформації до роботи

Завдання було розробити програму креслення гвинта згідно ДСТУ 10340-80 виконання 1 (рис. 3.1) для цього потрібно проаналізувати текст вихідних даних зі стандартів ДСТУ 10340-80.

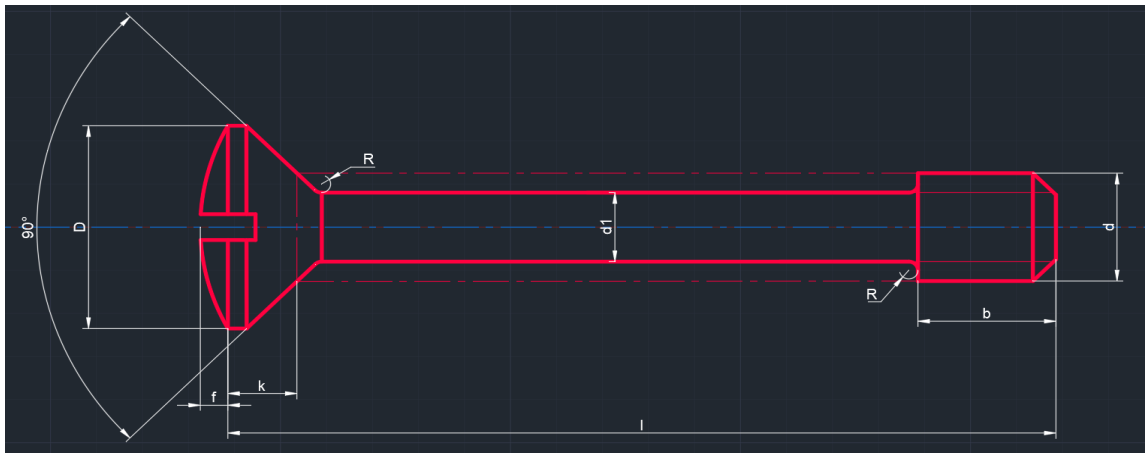


Рисунок 3.1 – Виконання 1 гвинта ДСТУ 10340-80

Дані було взято дані з таблиці 1 відповідно до виконання 1 з джерела [9]. Дані відображено в таблиці 3.1. Інші дані цієї таблиці не потрібні.

Таблиця 3.1 – Розміри гвинта, мм

Діаметр різьблення d	2,5	3	4	5	6	8	10	12
Ширина шліцу w^{*1}	0,60	0,80	1,00	1,20	1,60	2,00	2,50	3,00
Глибина шліцу h^{*1} напівпотайної головки	1,20	1,45	1,90	2,30	2,80	3,70	4,50	5,40
Діаметр стрижня d_1	1,6	2,0	2,8	3,5	4,0	5,5	7,0	9,0
Довжина різьблення b	3	4	5	6	8	10	12	16
Діаметр головки D	4,7	5,6	7,4	9,2	11,0	14,5	18,0	21,5
Висота головки (без сфери) k	1,50	1,65	2,20	2,50	3,00	4,00	5,00	6,00
Висота сфери f	0,60	0,75	1,00	1,25	1,50	2,00	2,50	3,00
Радіус під головкою R	0,2			0,4		0,5		0,6
Радіус сфери R_1	5,4	6,0	8,0	9,4	12,0	15,0	19,0	22,5
Діаметр плоского кінця dp^{*2}	1,5	2,0	2,5	3,5	4,0	5,5	7,0	8,5

Примітка 1. *1 – Додаткові дані про шліц з джерела [10].

Примітка 2. *2 – Додаткові дані про плоский кінець з джерела [11].

Виходячи з креслення гвинта згідно ДСТУ 10340-80 виконання 1 було взято додаткові дані про шліц з таблиці 2 зі стандартів ДСТУ 10340-80 (рис. 3.2) [10]. Дані відображено в таблиці 3.1. Інші дані цієї таблиці не потрібні.

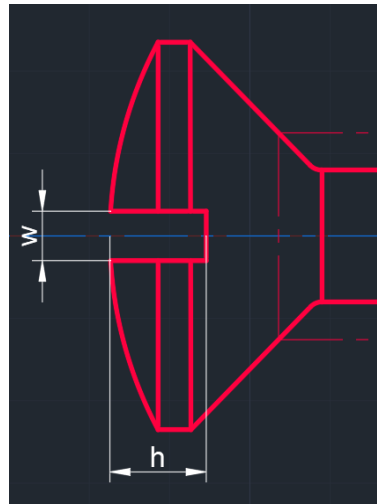


Рисунок 3.2 – Шліц гвинта ДСТУ 10340-80

Дані з джерел [9] та [10] об'єдно в одній таблиці 3.1.

Було взято дані щодо відношення діаметра різьблення до довжини гвинта дані з таблиці 2 зі стандартів ДСТУ 10340-80 [9] (табл. 3.2).

Проаналізувавши гвинт ДСТУ 10340-80 виконання 1 виявилось, що розміри фаски гвинта не вказано в стандарті ДСТУ. Дані про цю фаску було взято з джерела [11] (рис. 3.3) та дописано в таблицю 3.1.

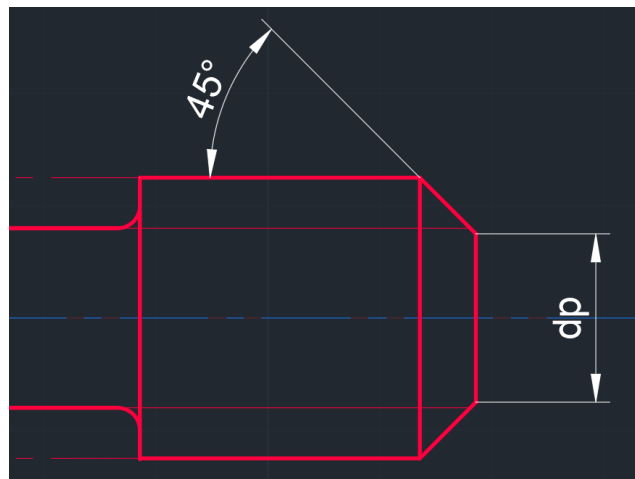


Рисунок 3.3 – Плоский кінець гвинта ДСТУ 10340-80

В процесі аналізу геометричних даних зображення конфліктують між собою значення f та $R1$. Тому в пріоритеті було обрано значення f а $R1$ буде нехтуватися від стандартів ДСТУ.

Таблиця 3.2 – Відношення діаметра різьблення до довжини гвинта

Довжина гвинта l , мм	Діаметр різьблення d , мм							
	2,5	3	4	5	6	8	10	12
6		#	-	-	-	-	-	-
8			#	-	-	-	-	-
10				-	-	-	-	-
12					#	-	-	-
(14)						-	-	-
16						-	-	-
(18)						-	-	-
20	-					-	-	-
(22)	-							-
25	-		Стандартні довжини					-
(28)	-							
32	-							
(36)	-							
40	-							
(45)	-							
50	-							
(55)	-							
60	-							
(70)	-	-	-					
80	-	-	-					

Під час креслення деякі діаметри різьблення не відповідають довжині гвинта: шліц відображається на різьбленні. Розміри з символом “#” в таблиці 3.2. Тому потрібно виключити комбінацію вхідних даних.

3.2 Схема алгоритму методики створення сервісної бібліотеки стандартних елементів

Для створення схеми алгоритму методики створення сервісної бібліотеки стандартних елементів було взято до уваги етапи які наведені на рисунку 3.4.



Рисунок 3.4 – Блок-схема алгоритму методики створення сервісної бібліотеки стандартних елементів

3.3 Розробка структури сервісного застосунку

Структурна схема бібліотеки наведена на рисунку 3.5.

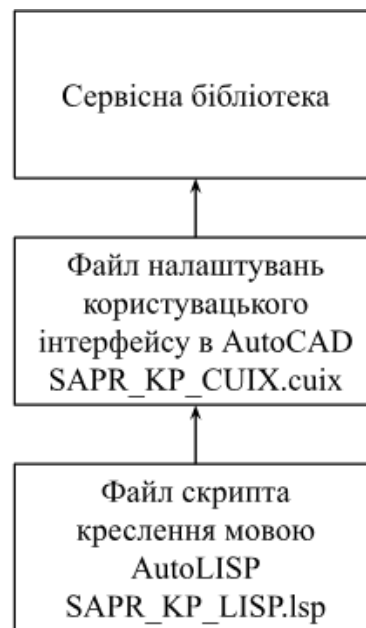


Рисунок 3.5 – Блок-схема структури сервісної бібліотеки стандартних елементів

3.4 Опис особливостей програмної реалізації бібліотеки

Для більш зручних та точних розрахунків потрібно проаналізувати початкові ескізи та виявити потрібну множину прив'язок, накреслити ескіз допоміжних точок (рис. 3.6) та обрати базову точку для кожного виду (точка pt1).

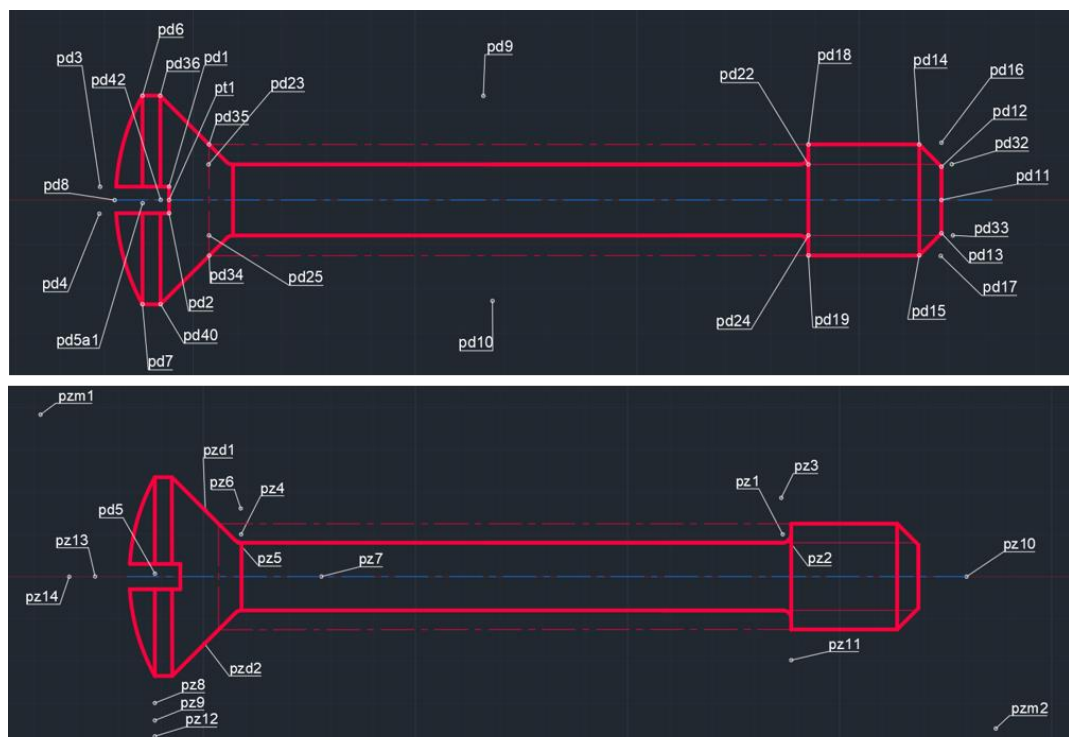


Рисунок 3.6 – Ескіз розташування допоміжних точок

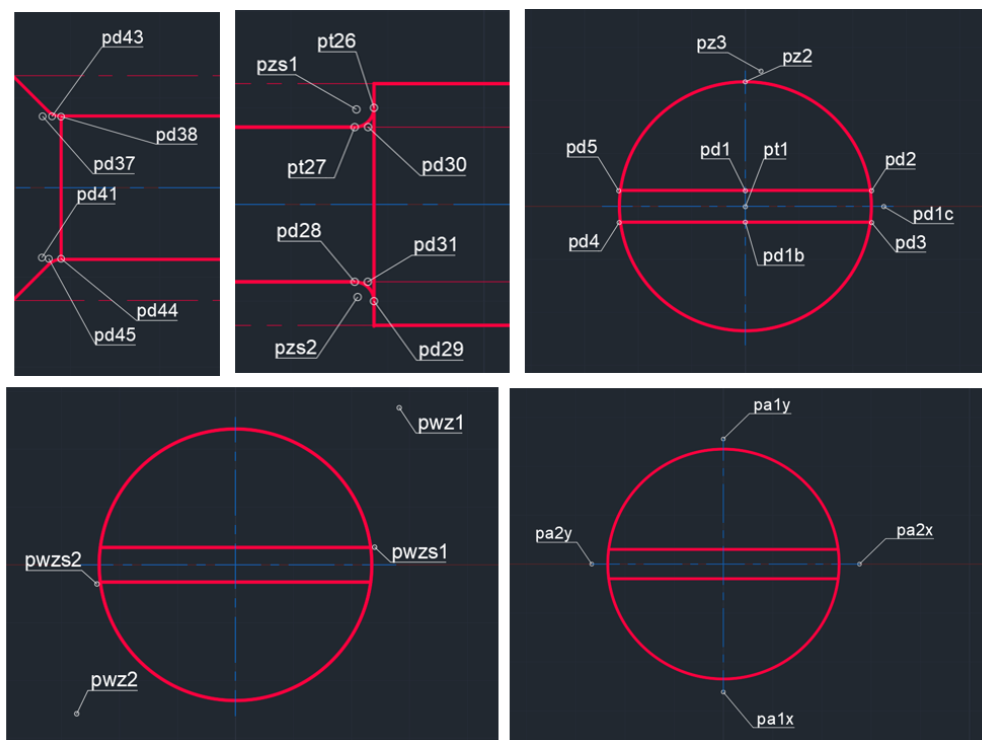


Рисунок 1а – Друга частина рисунку 3.6

Взаємодія між функціями між собою в програмному застосунку зображено на рисунку 3.7.

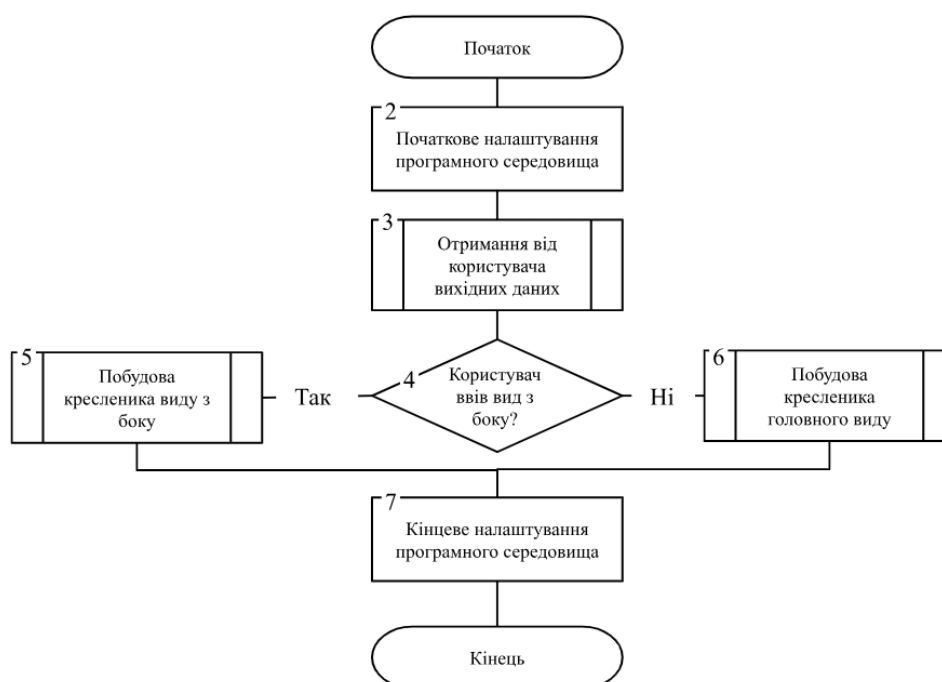


Рисунок 3.7 – Блок-схема головної функції програмного застосунку

В блоці 2 реалізована функція *tune_env* вона вимикає всі прив'язки і об'єкту прив'язку, встановлює розміщення тексту розмірів над лінією, довжин виступу розмірних ліній, розмір зони вибору та масштабу типів ліній, визначає змінні для користувацьких запитів введення імен шарів, створює шари з заданими властивостями. Команда *cmdecho* відповідає за виведення результатів виконання команд у вікно командного рядка.

В блоці 3 реалізовано функцію *user_input* вона оголошує списки, що містять параметри для різних діаметрів різьби, довжин гвинтів, список можливих видів креслення. Реалізує користувацьке введення базової точки, вибір виду креслення (головний/вид з боку), вибір діаметра різьби, вибір довжини гвинта (тільки для головного виду), введення кута повороту, проставлення розмірів та перевіряє на правильність введення користувачем всіх даних. В кінці формує список, що містить всі необхідні параметри для подальшого використання.

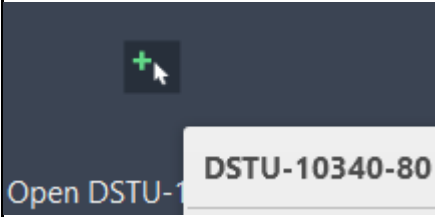
В блоці 5 реалізовано креслення виду з боку, в 6 блоці реалізовано креслення головного виду, а саме: визначення точок для позиціонування ліній та розмірів гвинта, створення геометричних об'єктів, якщо користувач вказав, що потрібно проставити розміри, виконується створення розмірів, якщо вказано кут повороту, гвинт повертається на заданий кут.

В блоці 7 оголошено команду *lwdisplay* відповідає за відображення шарів ліній в прозорих об'єктах у просторі моделі. Ця команда дозволяє побачити лінії, що знаходяться всередині або за прозорими об'єктами, забезпечуючи більш детальне і точне представлення моделі. Крім того, у цьому блоці оголошується команда *cmdecho* відповідає за виведення результатів виконання команд у вікно командного рядка.

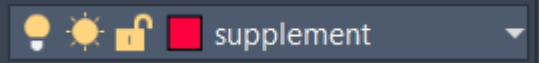
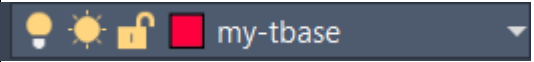
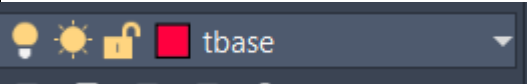
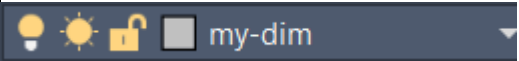
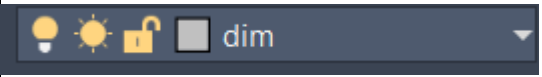
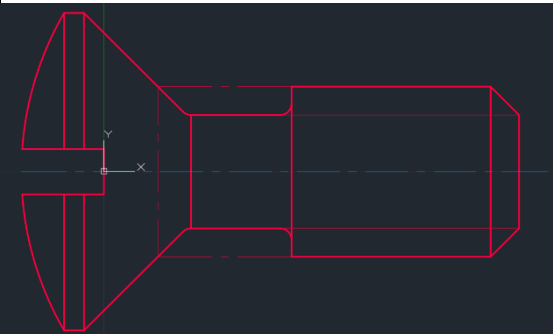
3.5 Порадник для користувача

Для коректного використання програми користувачем в таблиці 3.3 прописані кроки, правильні та неправильні введення в консолі, та результат кроку.


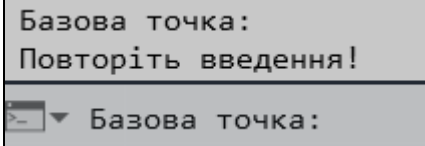
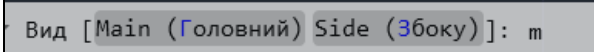
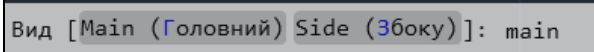
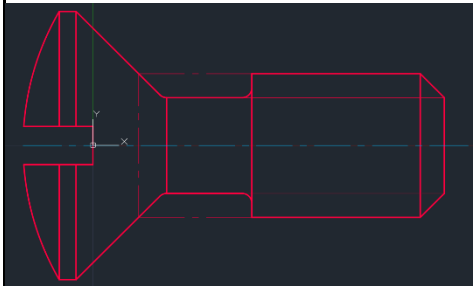
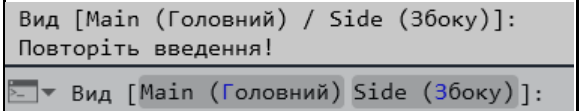
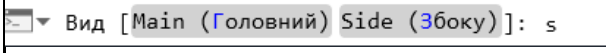
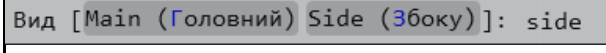
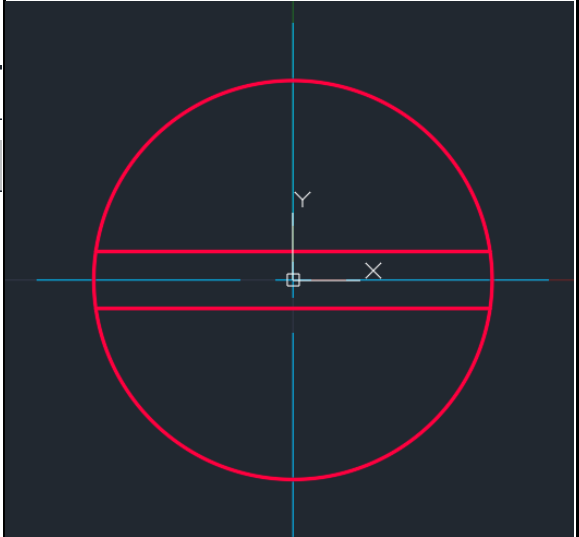
Таблиця 3.3 – Порадник для користувача

Етап №	Крок	Результат
1	<p>Натиснути кнопку лівою клавішею миші в вкладці DSTU робочого простору, в панелі Open DSTU-10340-80.</p> 	<p>Запуститься програмна реалізація.</p> 
2	<p>Ввести в полі консолі назву шару основних ліній, наприклад “my-base”. Натиснути кнопку ENTER.</p> 	<p>Створиться шар під назвою “my-base”.</p> 
	<p>Залишити порожнім. Натиснути кнопку ENTER</p> 	<p>Створиться шар за замовчуванням під назвою “base”.</p> 
3	<p>Ввести в полі консолі назву шару осьових ліній, наприклад “my-axial”. Натиснути кнопку ENTER.</p> 	<p>Створиться шар під назвою “my-axial”.</p> 
	<p>Залишити порожнім. Натиснути кнопку ENTER</p> 	<p>Створиться шар за замовчуванням під назвою “axial”.</p> 
4	<p>Ввести в полі консолі назву шару осьових ліній, наприклад “my-supp”. Натиснути кнопку ENTER.</p> 	<p>Створиться шар під назвою “my-supplement”.</p> 

Продовження таблиці 3.3

Етап №	Крок	Результат
4	Залишити порожнім. Натиснути кнопку ENTER <div>введіть ім'я шару додаткових ліній <supplement>:</div>	Створиться шар за замовчуванням під назвою “supplement”. 
5	Ввести в полі консолі назву шару тонких ліній, наприклад “my-tbase”. Натиснути кнопку ENTER. <div>Введіть ім'я шару тонких ліній <tbase>: my-tbase</div>	Створиться шар під назвою “my-tbase”. 
	Залишити порожнім. Натиснути кнопку ENTER <div>Введіть ім'я шару тонких ліній <tbase>:</div>	Створиться шар за замовчуванням під назвою “tbase”. 
6	Ввести в полі консолі назву шару розмірних ліній, наприклад “my-dim”. Натиснути кнопку ENTER. <div>Введіть ім'я шару розмірних ліній <dim>: my-dim</div>	Створиться шар під назвою “my-dim”. 
	Залишити порожнім. Натиснути кнопку ENTER <div>Введіть ім'я шару розмірних ліній <dim>:</div>	Створиться шар за замовчуванням під назвою “dim”. 
7	Ввести в полі консолі початкову точку, наприклад “0.0,0.0”. Натиснути кнопку ENTER. <div>Базова точка: 0.0,0.0</div>	Побудується кресленик гвинта відносно початкової точки 0.0,0.0. 

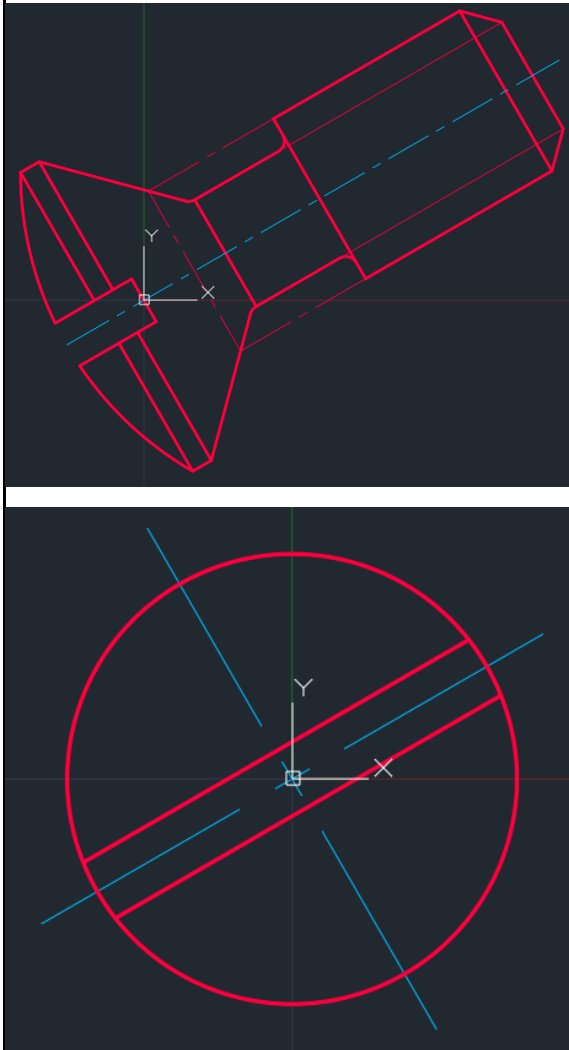
Продовження таблиці 3.3

Етап №	Крок	Результат
7	<p>Залишити порожнім.</p> <p>Натиснути кнопку ENTER.</p> 	<p>Програма виведе повідомлення, щоб повторити введення.</p> 
8	<p>Ввести вид гвинта, наприклад “m” або “main”.</p> <p>Натиснути кнопку ENTER.</p>  	<p>Побудується кресленик гвинта з головним видом.</p> 
	<p>Ввести невірне значення, наприклад “test”, “123” або залишити порожнім.</p> <p>Натиснути кнопку ENTER.</p>	<p>Програма виведе повідомлення, щоб повторити введення.</p> 
	<p>Ввести вид гвинта, наприклад “s” або “side”.</p> <p>Натиснути кнопку ENTER.</p>  	<p>Побудується кресленик гвинта вид збоку.</p> 

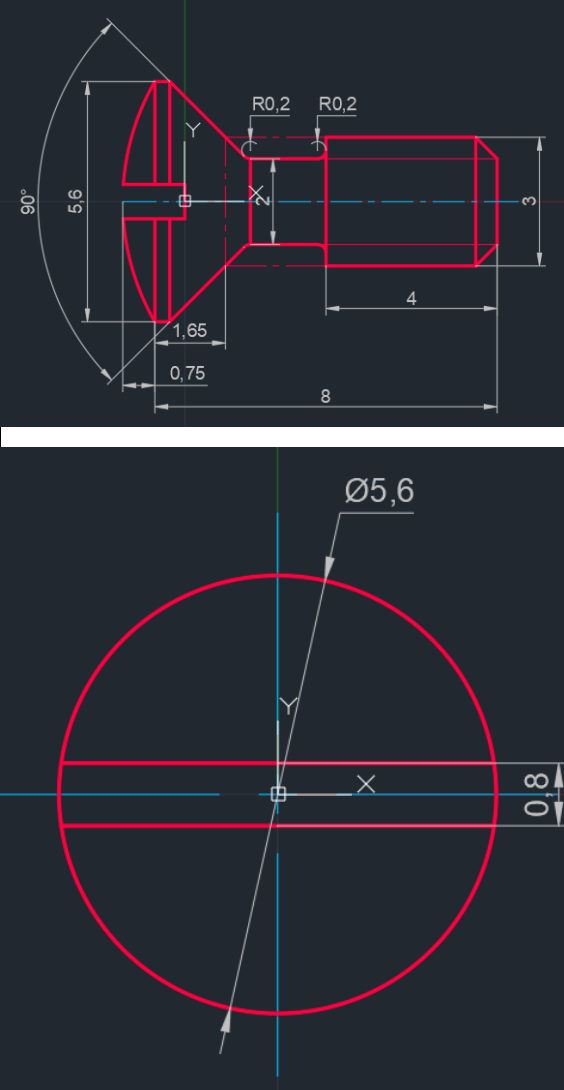
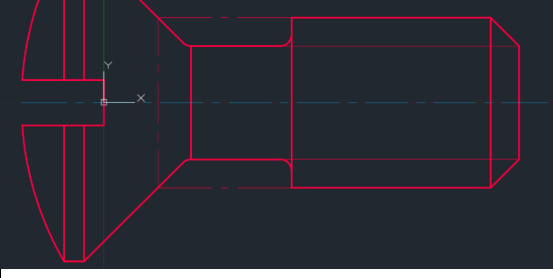
Продовження таблиці 3.3

Етап №	Крок	Результат
9	Ввести значення діаметр різблення, наприклад “3”. Натиснути кнопку ENTER.	Побудується кресленик гвинта з діаметром “3”.
	Діаметр різби: < 2.5 3 4 5 6 8 10 12 >: 3	
10	Ввести значення інше з запропонованого списку. Наприклад “123”, “test” або залишити пустим. Натиснути кнопку ENTER.	Програма виведе повідомлення, щоб повторити введення. Вид [Main (Головний) / Side (Збоку)]: side Діаметр різби: < 2.5 3 4 5 6 8 10 12 >: Повторіть введення!
	Діаметр різби:	▼ Діаметр різби:
10	Якщо на кроці 8 вказати значення для головного виду, програма відобразить текст про введення довжини гвинта за запропонованими значеннями до відповідного діаметру. Наприклад для гвинта з діаметром 3 ввести “8”. Натиснути кнопку ENTER.	Побудується кресленик гвинта з довжиною 8.
	Длина болта <8,10,12,14,16,18,20,22,25,28,32,36,40,45,50,55,60>: 8	
10	Якщо на кроці 8 вказати значення для головного виду, програма відобразить текст про введення довжини гвинта за запропонованими значеннями до відповідного	Програма виведе повідомлення, щоб повторити введення.
		Довжина гвинта <8,10,12,14,16,18,20,22,25,28,32,36,40,45,50,55,60>: Повторіть введення!

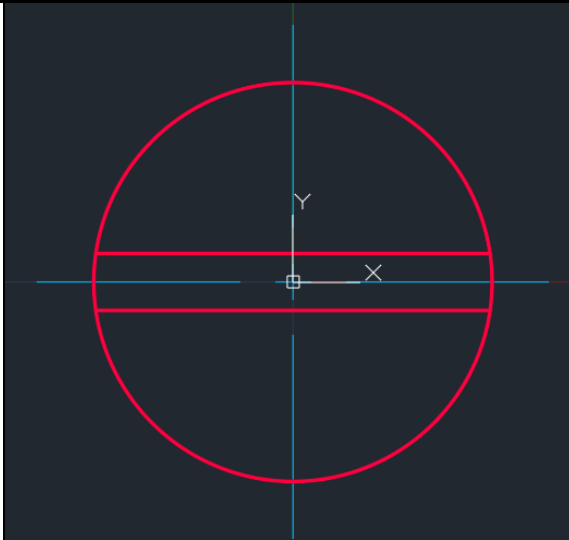
Продовження таблиці 3.3

Етап №	Крок	Результат
10	<p>діаметру. Ввести значення інше за запропоноване.</p> <p>Наприклад “123”, “test” або залишити порожнім.</p> <p>Натиснути кнопку ENTER.</p> <p>Длина болта <8,10,12,14,16,18,20,22,25,28,32,36,40,45,50,55,60>:</p>	
11	<p>Ввести значення куту повороту.</p> <p>Наприклад “30”.</p> <p>Натиснути кнопку ENTER.</p> <p>Кут повороту, градуси: 30</p>	<p>Побудується кресленик гвинта з кутом повороту 30 градусів.</p> 
12	Ввести значення для простановки розмірів. Наприклад “у” або “yes”.	Побудуються розміри до відповідного розміру кресленника

Продовження таблиці 3.3

Етап №	Крок	Результат
12	Натиснути кнопку ENTER.	Гвинта.
	<div data-bbox="312 443 911 517" data-label="Text">Проставити розміри (Yes/No): y</div>	
	<p>Ввести значення для простановки розмірів. Наприклад “n” або “no”.</p> <p>Натиснути кнопку ENTER.</p>	<p>Не побудуються розміри до відповідного розміру кресленника гвинта.</p>
	<div data-bbox="312 1816 911 1890" data-label="Text">Проставити розміри (Yes/No): no</div>	

Доповнення таблиці 3.3.

Етап №	Крок	Результат
12		

3.6 Порадник для програміста

3.6.1 Підключення директорію з програмним забезпеченням до конфігураційних налаштувань середовища розробки

Перед початком роботи потрібно створити робочу папку. Для цього в робочій директорії на порожньому місці натискаємо правою клавішею миші, в випадаючому меню обираємо пункт **Properties** (рис. 3.8).

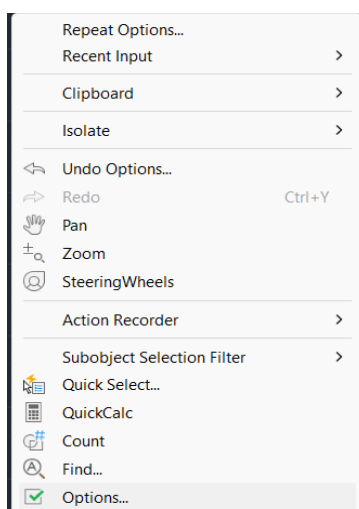


Рисунок 3.8 – Випадаюче меню в робочій директорії

В вікні **Options** в вкладці **Files** додати директорію з початковим програмним забезпечення в **Support File Search Path**. Для цього натиснути кнопку **Add** та вказати шлях до директорії (*D:\my-save-projects\SAPR_KP*) (рис. 3.9).

Таким самим чином вказати шлях до директорії з початковим кодом в **Trusted Locations** (рис. 3.10).

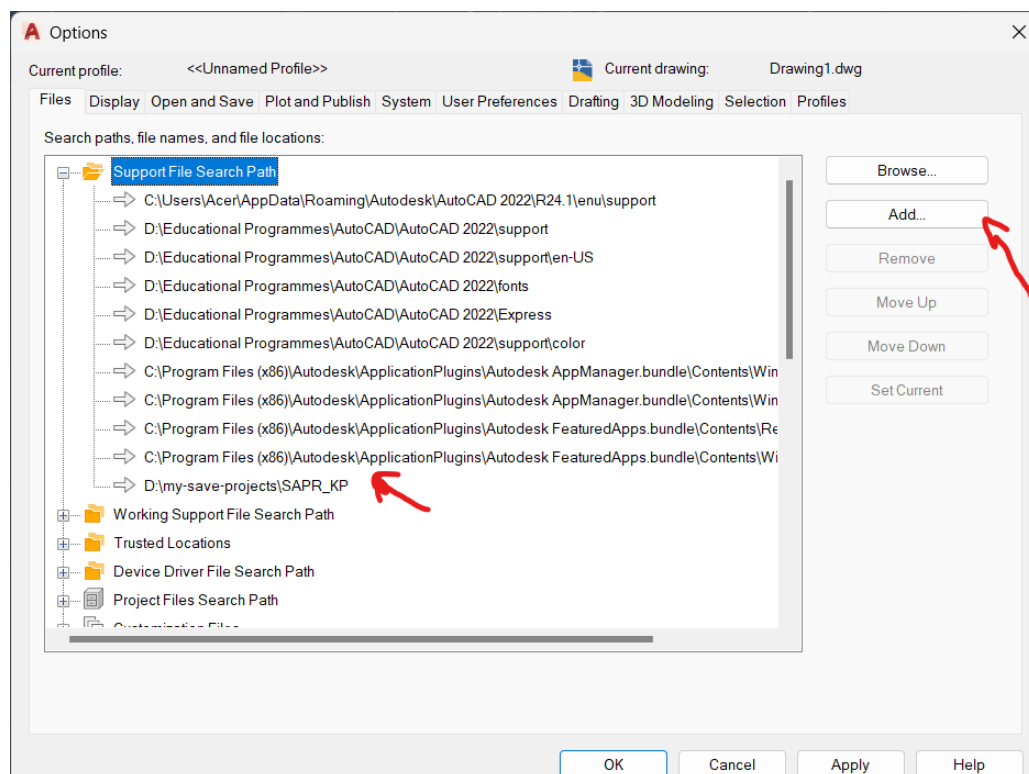


Рисунок 3.9 – Додавання директорії з програмним забезпеченням в **Support File Search Path**

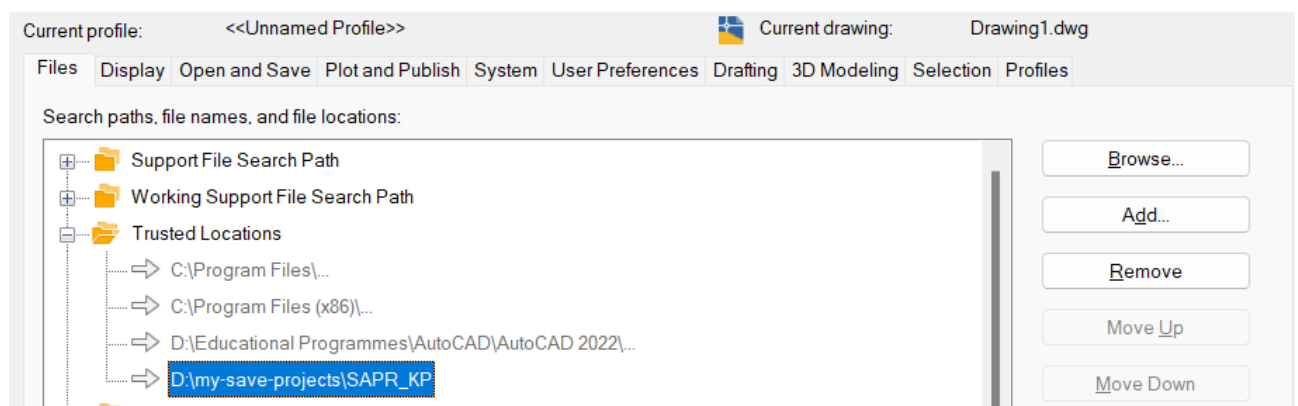


Рисунок 3.10 – Додавання директорії з програмним забезпеченням в **Trusted Locations**

Для збереження змін в вікні **Options** спочатку натиснути кнопку **Apply** а потім **ОК** в правому нижньому куті вікна (рис. 3.11).

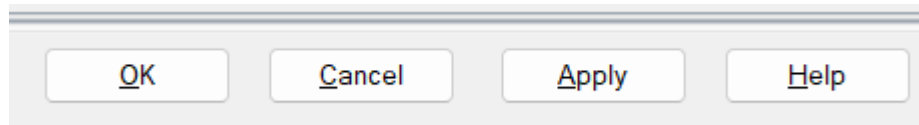



Рисунок 3.11 – Кнопки для збереження змін в вікні **Options**

3.6.2 Створення та підключення вкладки до робочого простору

Для створення панелі інструменту в середовищі AutoCAD для відкриття програмного застосунку потрібно натиснути на кнопку Workspace Switching  в правому нижньому куті в робочій директорії. З випадаючого меню натиснути **Customize**. Відкриється вікно **Customize User Interface** (рис. 3.12).

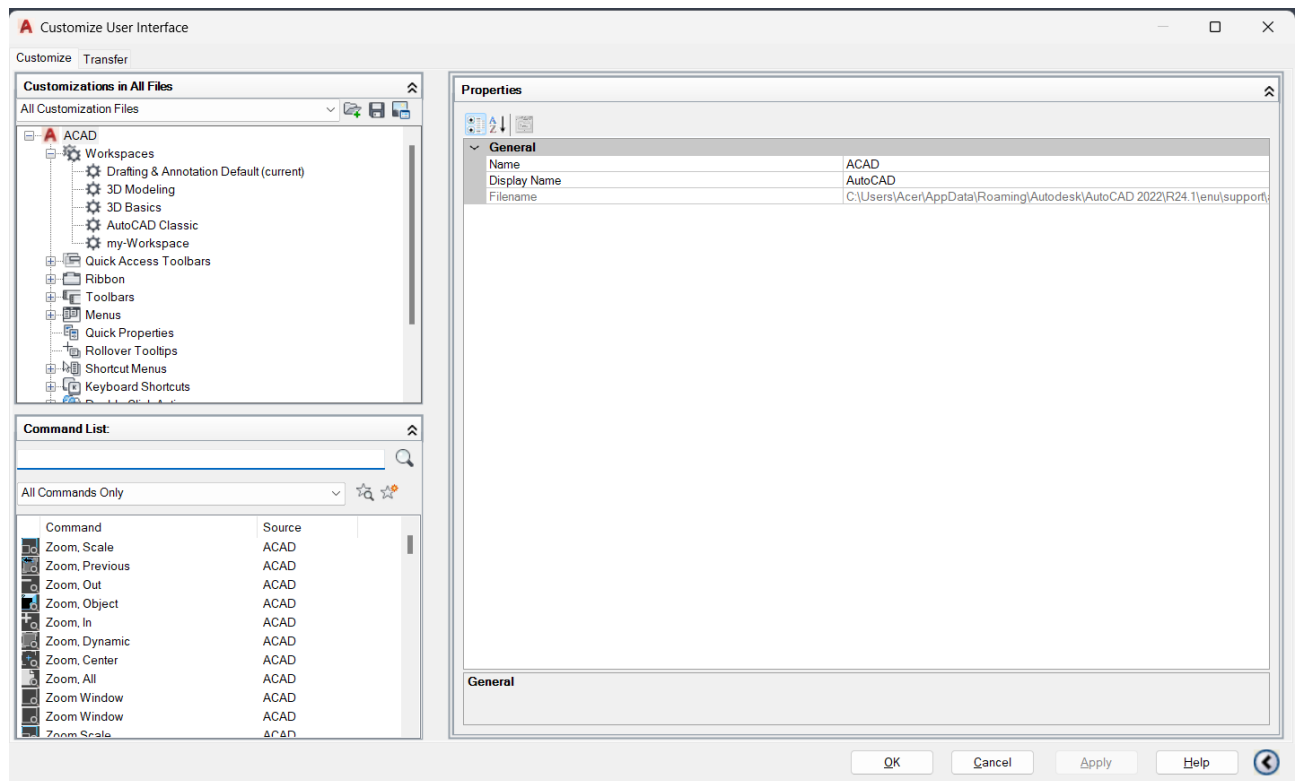



Рисунок 3.12 – Вікно **Customize User Interface**

Наступним кроком потрібно створити власну команду в секції **Command List** натиснути на кнопку **Create a new command** . В секції **Button Image** вибрати лаконічну картинку для кнопки (рис. 3.13).

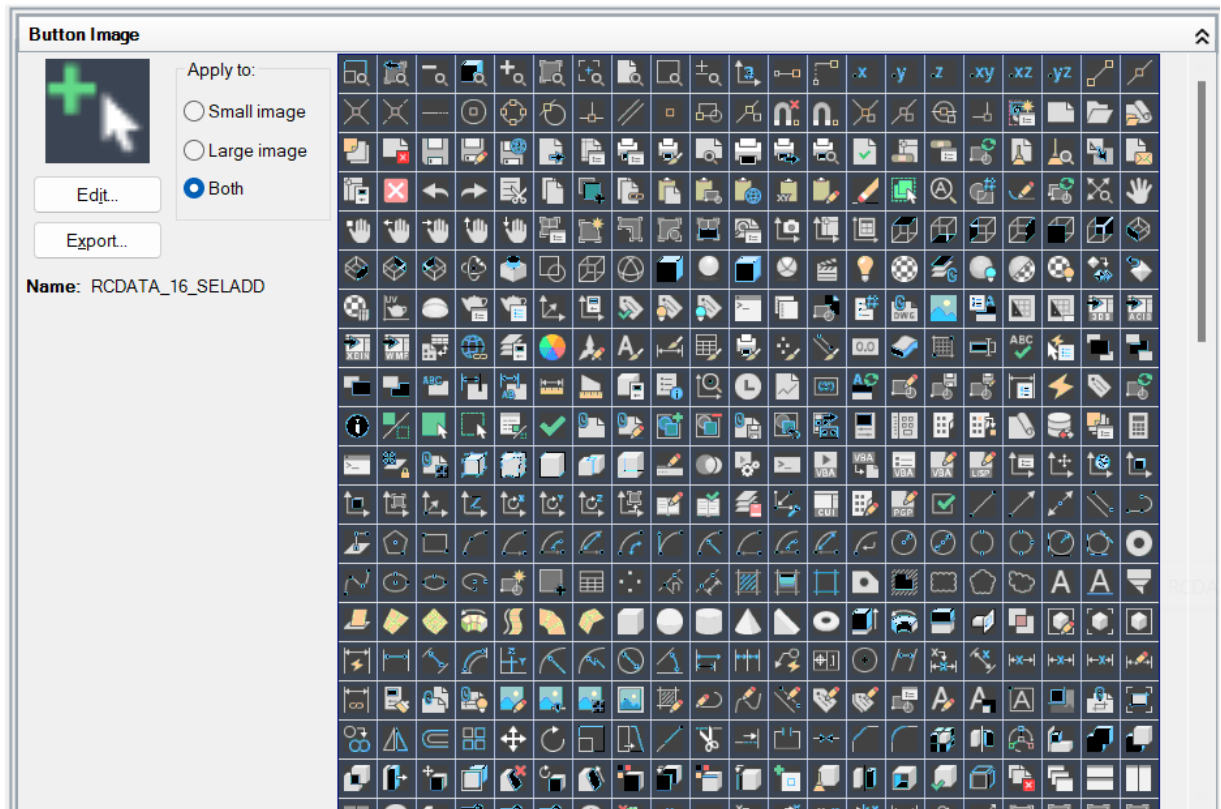


Рисунок 3.13 – Секція Button Image

В секції **Properties** в списку **Command** в пункті **Name** вказати назву кнопки (*DSTU-10340-80*), в пункті **Macro** записати макрос (*^C^C(load "SAPR_KP_LIST.lsp")(DSTU)*) з назвою файлу .lsp та головною функцією (рис 3.14).

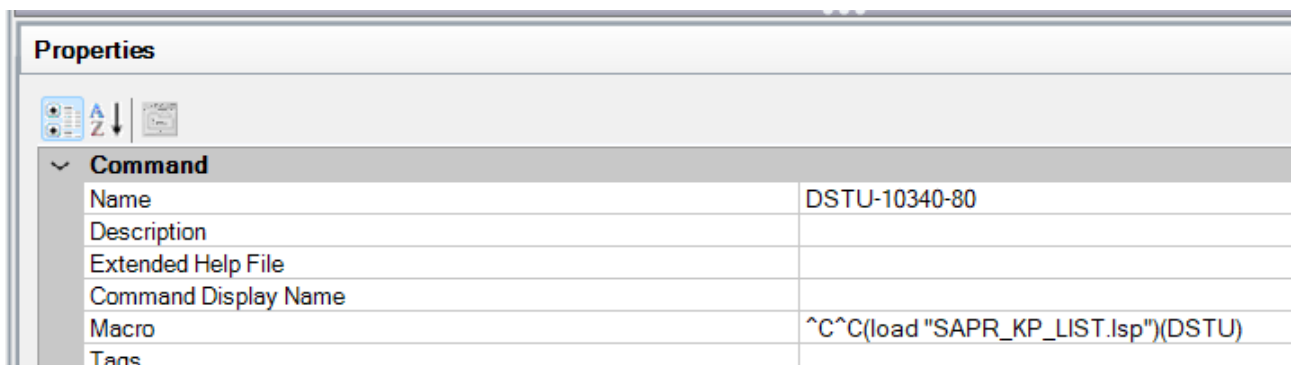


Рисунок 3.14 – Секція Properties

Для створення панелі в секції **Customizations in All Files** відкрити директорію **Ribbon/Panels**, клацнути правою кнопкою миші, з випадаючого меню натиснути **New Panels** (рис. 3.15).

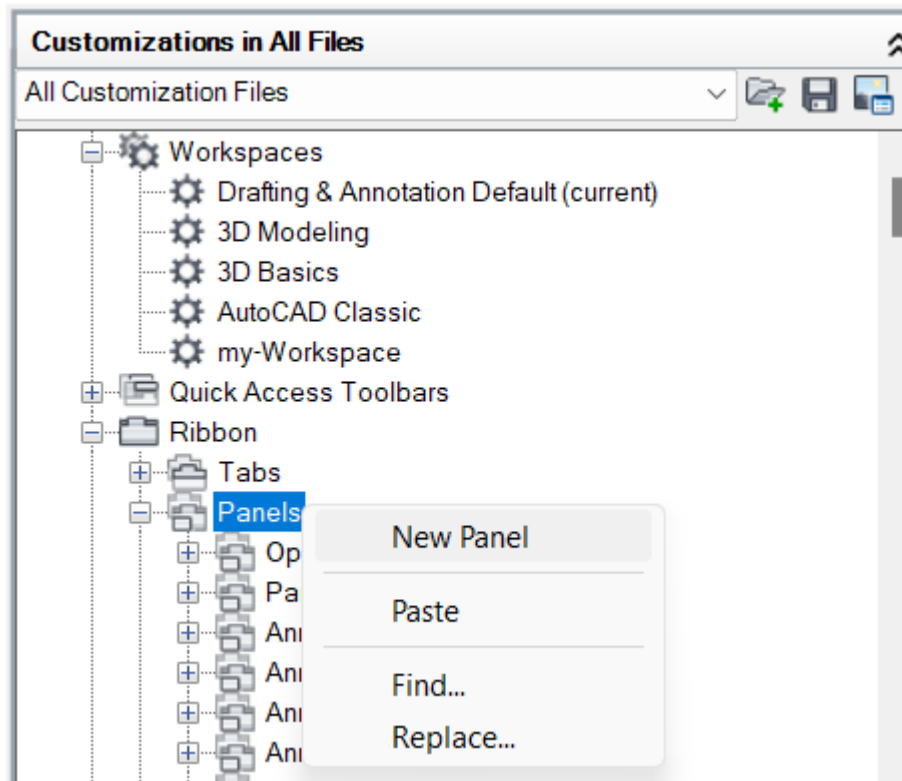


Рисунок 3.15 – Створення панелі

Виділити новостворену панель, натискаємо клавішу **F2** задати назву (*Open DSTU-10340-80*).

З секції **Command List** перетягнути створену команду в директорію **Open DSTU-10340-80/Row 1** (рис 3.16).

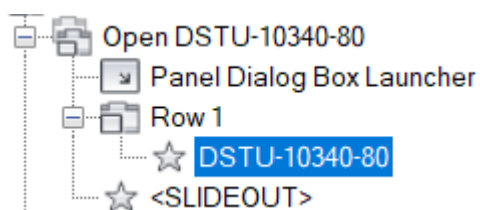


Рисунок 3.16 – Додавання команди в панель

Для створення вкладки в секції **Customizations in All Files** відкрити директорію **Ribbon/Tabs**, клацнути правою кнопкою миші, з випадаючого меню натиснути **New Tabs** (рис. 3.17).

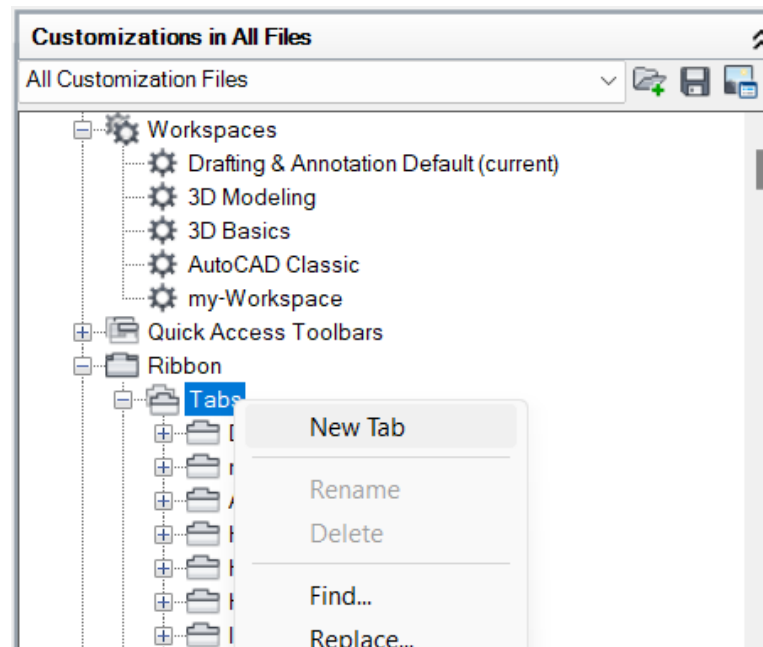


Рисунок 3.17 – Створення вкладки

Виділити новостворену панель, натискаємо клавішу **F2** задати назву (*DSTU-10340-80*).

Директорію **Ribbon/Panels/Open DSTU-10340-80** перетягнути в директорію **Ribbon/Tabs/DSTU-10340-80** (рис. 3.18).

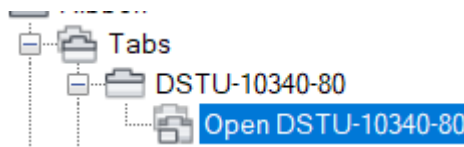


Рисунок 3.18 – Додавання панелі в вкладку

Вибрати **Workspace** в який потрібно додати новостворену вкладку (*Drafting & Annotation*).

Перетягнути директорію **Ribbon/Tabs/DSTU-10340-80** в директорію **Drafting & Annotation/Ribbon** в секції **Workspace Contents** (рис. 3.19).

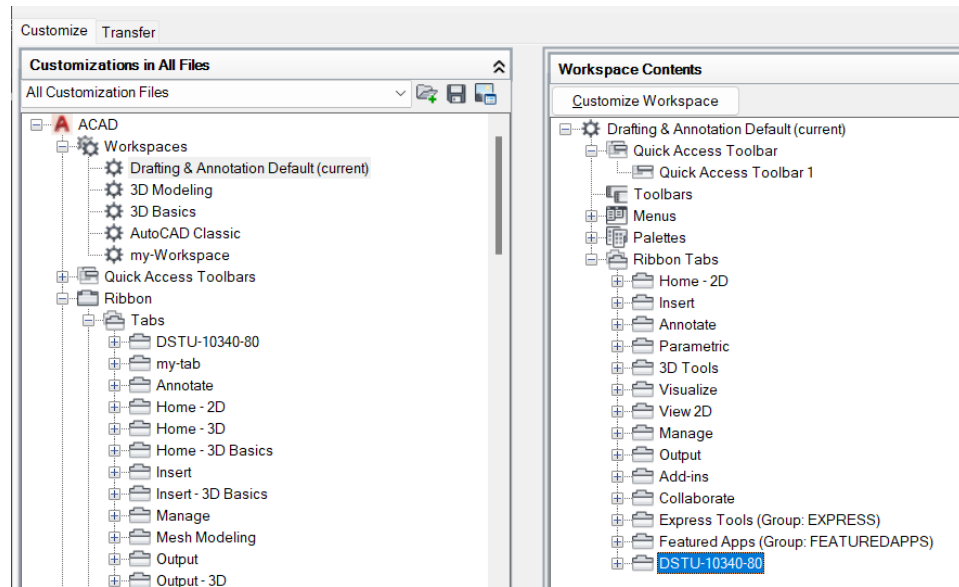


Рисунок 3.19 – Додавання вкладки **DSTU-10340-80** до робочого простору

Для збереження змін в вікні **Customize User Interface** спочатку натиснути кнопку **Apply** а потім **OK** в правому нижньому куті вікна (рис. 3.20).

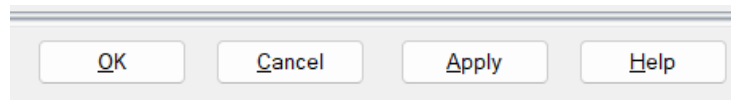


Рисунок 3.20 – Кнопки для збереження змін в вікні **Customize User Interface**

В результаті з'явиться додаткова вкладку в робочій директорії **Drafting & Annotation** (рис 3.21).

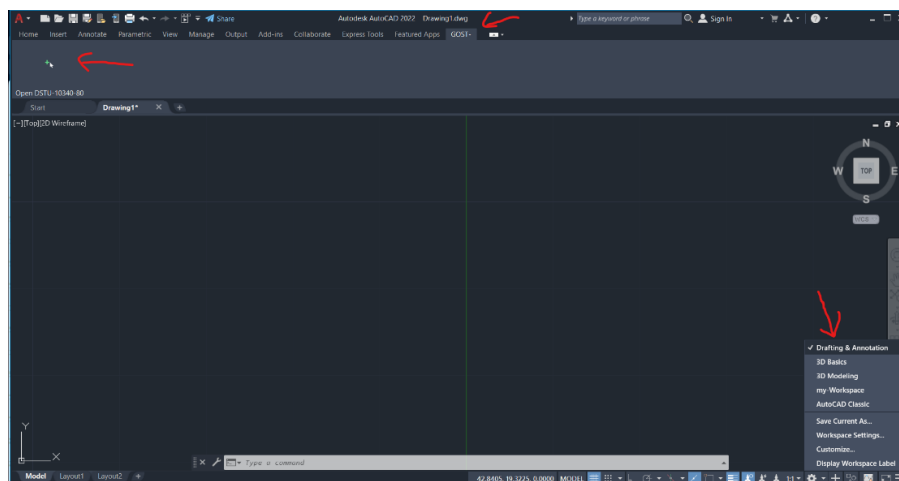


Рисунок 3.21 – Результат додавання вкладки в робочу директорію

3.6.3 Порадник для підтримки програмного застосунку

Для зміни та коригування програмної реалізації сервісної бібліотеки AutoCAD гвинт невинтаючий з напівпотайною головкою ДСТУ 10340-80 потрібно звернути увагу на елементи, наведені в таблиці 3.4.

Таблиця 3.4 – Перелік елементів коду для підтримки програми

Елемент коду	Опис
Список <i>d_other</i>	Оголошення табличних даних для розміра гвинта (табл 3.1). Це список списків: <ul style="list-style-type: none"> – нульовий елемент (d): діаметр різьблення; – перший (w): ширина шліцу; – другий (h): глибина шліцу напівпотайної головки; – третій (d1): діаметр стрижня; – четвертий (b): довжина різьблення; – п'ятий (D): діаметр головки; – шостий (k): висота головки (без сфери); – сьомий (f): висота сфери; – восьмий (R): радіус під головкою; – дев'ятий (dp): діаметр плоского кінця.
Список <i>L_all</i>	Оголошення всіх розмірів довжини гвинта (табл 3.2).
Список <i>L_unrec</i>	Оголошення не рекомендованих розмірів довжини гвинта. В таблиці 3.2 ці значення обгорнуті в круглі дужки.
Список <i>d_Lb</i>	Оголошення табличних даних відношення діаметра різьблення до довжини гвинта (табл. 3.2). Це список списків, шаблоном є вираз (a (b c)): <ul style="list-style-type: none"> – перше значення (a): діаметр різьблення; – друге (b): мінімальна дозволена довжина гвинта; – третє (c): максимальна дозволена довжина гвинта.
Список <i>d_all</i>	Оголошення всіх розмірів для діаметра гвинта (табл. 3.2)
Функція <i>tune_env</i>	Для зміни налаштування програмного середовища та для змінення параметрів шарів.
Функція <i>plot_main</i>	Для виправлення або дописування опорних точок, змінення креслення точок, проставлення розмірів головного виду
Функція <i>plot_said</i>	Для виправлення або дописування опорних точок, змінення креслення точок, проставлення розмірів виду з боку
Функція <i>DSTU</i>	Для введення значень користувача та виведення кресленика

ВИСНОВКИ

В даній курсовій роботі було реалізовано програмну реалізацію сервісних бібліотек AutoCAD невинпадаючого гвинта з напівпотайною головою ДСТУ 10340-80.

Цю бібліотеку можна підключити в вигляді кнопки до свого робочого простору.

Програма реалізує такі дії, як:

- оголошення вихідних даних;
- перевірка відповідних розмірів довжини гвинта до діаметра різьблення;
- налаштування шарів за замовчуванням (колір, тип, товщина, назва);
- налаштування середовища розробки (відключення всіх прив'язок під час креслення, налаштування масштабування типів ліній, налаштування для простановки розмірів);

- розрахування опорних точок для побудови кресленика;
- креслення фігур за заданими опорними точками;
- реалізує кут повороту кресленика;
- отримує від користувача дані.

При запуску програма отримує від користувача такі дані, як:

- назву шарів для побудови;
- початкову точку;
- вид креслення;
- розмір діаметра різьблення;
- при введенні головного виду ще потрібно ввести довжину гвинта;
- кут повороту кресленика та запит на простановку розміру.

При введенні некоректного значення програма повідомить про помилку та знову запитає про введення даних.

Як результат введення коректних даних будується відповідний кресленик.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кветний, Р.Н. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 [Електрон. ресурс] / Р.Н. Кветний, І.В. Богач, О.Р. Бойко, О.Ю. Сифина, О.М. Шушура. – Режим доступу: https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp%27yuterne_modelyuvannya_system_procesiv/t1/173.htm.
2. САПР об'єктів малої енергетики / Під редакцією В. Г. Сліпченка – К.: Знання України, 2007. -216 с. Бібліогр.: с. 213-214.
3. Норенкова, І.П. Системи автоматизованого проектування: Підручник для вузів: У 9 книжках / І.П. Норенкова – М.: Вищ. шк. 1986.
4. Норенков, І.П. Інформаційна підтримка наукомісткої продукції. CALS-технології / І.П. Норенков, П.К. Кузьмін – Москва: МГТУ ім. Баумана, 2002.
5. Колчин, О.Ф. Управління життєвим циклом продукції / О.Ф. Колчин, М.В. Овсянников, О.Ф. Стрекалов, С.В. Сумароков – М.: Анахарсис, 2002.
6. Gladfelter, D. 8 чудових функцій AutoCAD, які цінують найкращі менеджери САПР [Електрон. ресурс] / D. Gladfelter. – Режим доступу: <https://wiseit.com.ua/8-autocad-features/>.
7. McCarthy, J. Lisp [Електрон. ресурс] / J. McCarthy. – Режим доступу: https://uk.wikipedia.org/wiki/Lisp#Формальний_опис_мови.
8. Щербина, В.Ю. Програмування на функціональній мові AutoLISP при проектуванні технологічного обладнання / В.Ю. Щербина, О.С. Сахаров, О.В. Гондляр, В.І. Сівецький – Київ: НТУУ «КПІ», 2014, с. 7 - 16.
9. Гвинти з напівтайною головкою НЕВИПАДНІ КЛАСУ ТОЧНОСТІ [Електроний ресурс] / Постановою Державного комітету СРСР за стандартами від 7 липня 1980 р. № 3429 – Режим доступу: <https://meganorm.ru.translate.goog/Data2/1/4294840/4294840352.htm>.
10. Шліци прямі для гвинтів і шуптів / Офіційне видання – Москва: Стандартні форми, 2016, с. 2 - 3.

11. Кінці болтів, гвинтів і шпильок / Офіційне видання – Мінск: Міждержавна рада зі стандартизації, метрології та сертифікації, с. 4.

ДОДАТОК А
Текст програми

```

(setq ui_err "\n Повторіть введення!"
  yn_list (list "y" "yes" "n" "no")
)

(defun user_input ()

(setq ; d w h d1 b D k f R dp
  d_other (list
    '(2.5 0.60 1.20 1.6 3 4.7 1.50 0.60 0.2 1.5)
    '(3 0.80 1.45 2.0 4 5.6 1.65 0.75 0.2 2.0)
    '(4 1.00 1.90 2.8 5 7.4 2.20 1.00 0.2 2.5)
    '(5 1.20 2.30 3.5 6 9.2 2.50 1.25 0.4 3.5)
    '(6 1.60 2.80 4.0 8 11.0 3.00 1.50 0.4 4.0)
    '(8 2.00 3.70 5.5 10 14.5 4.00 2.00 0.5 5.5)
    '(10 2.50 4.50 7.0 12 18.0 5.00 2.50 0.5 7.0)
    '(12 3.00 5.40 9.0 16 21.5 6.00 3.00 0.6 8.5)
  )
  L_all (list 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 25.0 28.0 32.0
    36.0 40.0 45.0 50.0 55.0 60.0 70.0 80.0
  )
  L_unrec (list 14.0 18.0 22.0 28.0 36.0 45.0 55.0 70.0)
  d_Lb (list
    '(2.5 (6.0 18.0))
    '(3.0 (8.0 60.0))
    '(4.0 (10.0 60.0))
    '(5.0 (12.0 80.0))
    '(6.0 (14.0 80.0))
    '(8.0 (22.0 80.0))
    '(10.0 (22.0 80.0))
    '(12.0 (28.0 80.0))
  )
  d_all (list 2.5 3.0 4.0 5.0 6.0 8.0 10.0 12.0)
  base_point (list 0.0 0.0)
  view "M"
  d (caar d_Lb)
  views (list "m" "main" "s" "side")
  ui_base "\n Базова точка: "
  ui_view "\n Вид [Main (Головний) / Side (Збоку)]: "
  ui_dia "\n Діаметр різьби: "
  ui_unrec "\n Значення не рекомендується!"
  ui_len "\n Довжина гвинта"
  ui_ang "\n Кут повороту, градуси: "
  ui_dim "\n Проставити розміри (Yes/No): "

```

```

L_d    nil
L_ds    ""
a_rot    0.0
putdim    "y"
req_par    nil
aux_par    nil
runout    "n"
)

(setq base_point (getpoint ui_base))

(while (= base_point nil)
  (prompt ui_err)
  (setq base_point (getpoint ui_base))
)

(setq view (strcase (getstring ui_view) T))

(while (= (member view views) nil)
  (prompt ui_err)
  (setq view (strcase (getstring ui_view) T))
)

(setq d (getreal
  (strcat ui_dia
    " < "
    (apply 'strcat
      (mapcar '(lambda (x) (strcat (rtos x) " ")) d_all)
    )
    ">: "
  )
)
)

(while (= (member d d_all) nil)
  (prompt ui_err)
  (setq d (getreal ui_dia))
)

(if (/= (member d d_unrec) nil)
  (prompt ui_unrec)
)

```

```

(if (or (= view "m") (= view "main")))
  (progn
    (setq lims (cdr (assoc d d_Lb)))

    (foreach Li L_all
      (if (and (>= Li (caar lims)) (<= Li (cadr (last lims))))
        (progn
          (setq L_d (append L_d (list Li)))
          (setq L_ds (strcat L_ds (rtos Li 2 2) ","))
        )
      )
    )
  )

  (setq L (getreal
    (strcat ui_len
      " <"
      (substr L_ds 1 (- (strlen L_ds) 1))
      ">: "
    )
  )
  )

  (while (= (member L L_d) nil)
    (prompt ui_err)
    (setq L (getreal (strcat ui_len ": ")))
  )

  (if (/= (member L L_unrec) nil)
    (prompt ui_unrec)
  )
)

(setq other (assoc d d_other))
(setq a_rot (getreal ui_ang))
(while (not (numberp a_rot))
  (prompt ui_err)
  (setq a_rot (getreal ui_ang))
)

(setq putdim (strcase (getstring ui_dim) T))
(while (= (member putdim yn_list) nil)
  (prompt ui_err)
  (setq putdim (strcase (getstring ui_dim) T))
)

```



```

    ui_lax "\n Введіть ім'я шару осьових ліній"
    ui_ltb "\n Введіть ім'я шару тонких ліній"
    ui_ldm "\n Введіть ім'я шару розмірних ліній"
    ui_sup "\n введіть ім'я шару додаткових ліній"
)
(setq layer_base (olayer "240" "continuous" "0.30" ui_lb "base")
  layer_axial (olayer "140" "center" "0.15" ui_lax "axial")
  layer_sup (olayer "240" "center" "0.15" ui_sup "supplement")
  layer_thin (olayer "240" "continuous" "0.15" ui_ltb "tbase")
  layer_dims (olayer "9" "continuous" "0.15" ui_ldm "dim")
)
)

(defun tan (num)
  (/ (sin num) (cos num))
)

(defun plot_main (pd)
  (setq pt1 (list (car pd) (cadr pd))
    pd1 (list (car pd)
      (+ (/ (nth 1 other) 2.0) (cadr pd))
    )
    pd2 (list (car pd)
      (+ (- (/ (nth 1 other) 2.0)) (cadr pd))
    )
    pd3 (list (+ (- (- (nth 2 other)) 1) (car pd))
      (+ (/ (nth 1 other) 2.0) (cadr pd))
    )
    pd4 (list (+ (- (- (nth 2 other)) 1) (car pd))
      (+ (- (/ (nth 1 other) 2.0)) (cadr pd))
    )
    pd5 (list (+ (+ (- (nth 2 other)) (nth 7 other)) (car pd))
      (- (cadr pd) 0.1)
    )
    pd5a1 (list (car pd5)
      (+ (cadr pd) 0.1)
    )
    pd6 (list (+ (+ (- (nth 2 other)) (nth 7 other)) (car pd))
      (+ (/ (nth 5 other) 2.0) (cadr pd))
    )
    pd7 (list (car pd6)
      (- (cadr pd) (/ (nth 5 other) 2.0))
    )
  )
)

```

```

pd8 (list (+ (- (nth 2 other)) (car pd))
          (cadr pd)
        )
pd9 (list
      (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (/ (nth 7 req_par) 2.0))
        (car pd)
      )
      (+ (/ (nth 5 other) 2.0) (cadr pd))
    )
pd10 (list
       (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (/ (nth 7 req_par) 2.0))
         (car pd)
       )
       (+ (- (/ (nth 5 other) 2.0)) (cadr pd))
     )
pd11 (list
       (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
         (car pd)
       )
       (cadr pd)
     )
pd12 (list
       (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
         (car pd)
       )
       (+ (/ (nth 9 other) 2.0) (cadr pd))
     )
pd13 (list
       (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
         (car pd)
       )
       (+ (- (/ (nth 9 other) 2.0)) (cadr pd))
     )
pd14 (list
       (+
         (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
           (- (/ (nth 0 other) 2.0) (/ (nth 9 other) 2.0))
         )
       (car pd)
     )
     (+ (/ (nth 0 other) 2.0) (cadr pd))
   )
pd15 (list

```

```

(+
  (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
    (- (- (/ (nth 9 other) 2.0)) (- (/ (nth 0 other) 2.0)))
  )
  (car pd)
)
(+ (- (/ (nth 0 other) 2.0)) (cadr pd))
)
pd16 (list
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
    (car pd)
  )
  (+ (/ (nth 0 other) 2.0) (cadr pd))
)
pd17 (list
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
    (car pd)
  )
  (+ (- (/ (nth 0 other) 2.0)) (cadr pd))
)
pd18 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
    (car pd)
  )
  (+ (/ (nth 0 other) 2.0) (cadr pd))
)
pd19 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
    (car pd)
  )
  (+ (- (/ (nth 0 other) 2.0)) (cadr pd))
)
pd22 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )

```

```

      (car pd)
    )
    (+ (/ (nth 3 other) 2.0) (cadr pd))
  )
pd23 (list
      (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other)) (car pd))
      (+ (/ (nth 3 other) 2.0) (cadr pd))
    )
pd24 (list
      (+
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (car pd)
    )
    (+ (- (/ (nth 3 other) 2.0)) (cadr pd))
  )
pd25 (list
      (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other)) (car pd))
      (+ (- (/ (nth 3 other) 2.0)) (cadr pd))
    )
pd26 (list
      (+
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (car pd)
    )
    (+ (+ (/ (nth 3 other) 2.0) (nth 8 other)) (cadr pd))
  )
pd27 (list
      (+
        (-
          (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
            (nth 4 other)
          )
        (nth 8 other)
      )
      (car pd)
    )
    (+ (/ (nth 3 other) 2.0) (cadr pd))
  )
pd28 (list

```

```

(+
  (-
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
    (nth 8 other)
  )
  (car pd)
)
(+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)
pd29 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
    (car pd)
  )
  (+ (- (- (/ (nth 3 other) 2.0)) (nth 8 other)) (cadr pd))
)
pd30 (list
  (+
    (-
      (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
        (nth 4 other)
      )
      (/ (nth 8 other) 2.0)
    )
    (car pd)
  )
  (+ (/ (nth 3 other) 2.0) (cadr pd))
)
pd31 (list
  (+
    (-
      (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
        (nth 4 other)
      )
      (/ (nth 8 other) 2.0)
    )
    (car pd)
  )
  (+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)

```

```

)
pzs1 (list
  (+
    (-
      (-
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (/ (nth 8 other) 2.0)
    )
    0.1
  )
  (car pd)
)
(+ (+ (/ (nth 3 other) 2.0) 0.1) (cadr pd))
)
pzs2 (list
  (+
    (+
      (-
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (/ (nth 8 other) 2.0)
    )
    0.1
  )
  (car pd)
)
(+ (- (- (/ (nth 3 other) 2.0)) 0.1) (cadr pd))
)
pd32 (list
  (+
    (+
      (-
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (nth 8 other)
    )
    (* (nth 4 other) 1.3)
  )
  (car pd)
)

```

```

    )
    (+ (/ (nth 3 other) 2.0) (cadr pd))
  )
pd33 (list
  (+
    (+
      (-
        (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
          (nth 4 other)
        )
      (nth 8 other)
    )
    (* (nth 4 other) 1.3)
  )
  (car pd)
)
(+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)
pd34 (list
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other)) (car pd))
  (+ (- (/ (nth 0 other) 2.0)) (cadr pd))
)
pd35 (list
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other)) (car pd))
  (+ (/ (nth 0 other) 2.0) (cadr pd))
)
pd36 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (/ (nth 5 other) 2.0) (/ (nth 0 other) 2.0))
    )
    (car pd)
  )
  (+ (/ (nth 5 other) 2.0) (cadr pd))
)
pd37 (list
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
    (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
    (car pd)
  )
  (+ (/ (nth 3 other) 2.0) (cadr pd))
)
pd38 (list

```



```

(+
  (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
    (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
    (nth 8 other)
  )
  (car pd)
)
(+ (/ (nth 3 other) 2.0) (cadr pd))
)
pd40 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (- (/ (nth 0 other) 2.0)) (- (/ (nth 5 other) 2.0)))
    )
    (car pd)
  )
  (+ (- (/ (nth 5 other) 2.0)) (cadr pd))
)
pd41 (list
  (+
    (+ (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (- (/ (nth 3 other) 2.0)) (- (/ (nth 0 other) 2.0)))
    )
    (car pd)
  )
  (+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)
pd42 (list
  (+
    (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (- (/ (nth 0 other) 2.0)) (- (/ (nth 5 other) 2.0)))
    )
    (car pd)
  )
  (cadr pd)
)
pd43 (list
  (+
    (+ (/ (nth 8 other) (tan (* 0.375 pi)))
      (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
    )
    (car pd)
  )

```

```

)
(+ (/ (nth 3 other) 2.0) (cadr pd))
)
pd44 (list
  (+
    (+
      (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
      (- (- (/ (nth 3 other) 2.0)) (- (/ (nth 0 other) 2.0)))
    )
    (nth 8 other)
  )
  (car pd)
)
(+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)
pd45 (list
  (+
    (+ (/ (nth 8 other) (tan (* 0.375 pi))))
    (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
    (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
  )
  (car pd)
)
(+ (- (/ (nth 3 other) 2.0)) (cadr pd))
)
pz1 (list
  (+
    (-
      (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
    (nth 8 other)
  )
  (car pd)
)
(+ (+ (/ (nth 3 other) 2.0) (nth 8 other)) (cadr pd))
)
pz2 (polar pz1 (/ (* 7 pi) 4) (nth 8 other))
pz3 (list
  (+
    (-
      (- (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 7 req_par))
      (nth 4 other)
    )
  )

```

```

    )
    (nth 8 other)
  )
  (car pd)
)
(+ (+ (/ (nth 3 other) 2.0) (* (nth 8 other) 4)) (cadr pd))
)
pz4 (list
  (+
    (+ (/ (nth 8 other) (tan (* 0.375 pi))))
    (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
    (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
  )
  (car pd)
)
(+ (+ (/ (nth 3 other) 2.0) (nth 8 other)) (cadr pd))
)
pz5 (polar pz4 (/ (* 13 pi) 9) (nth 8 other))
pz6 (list
  (+
    (+ (/ (nth 8 other) (tan (* 0.375 pi))))
    (+ (+ (- (nth 2 other)) (nth 7 other)) (nth 6 other))
    (- (/ (nth 0 other) 2.0) (/ (nth 3 other) 2.0))
  )
  (car pd)
)
(+ (+ (/ (nth 3 other) 2.0) (* (nth 8 other) 4)) (cadr pd))
)
pzb1 (list (+ (car pd37) (* (nth 8 other) 3))
  (+ (cadr pd37) (* (nth 8 other) 3))
)
pzb2 (list (- (car pd37) (* (nth 8 other) 3))
  (- (cadr pd37) (* (nth 8 other) 3))
)
pz7 (list (+ (car pd) (* (nth 6 other) 1.25))
  (cadr pd)
)
pz8 (list (car pd7)
  (- (cadr pd7) (* (nth 6 other) 0.3))
)
pz9 (list (car pd7)
  (- (cadr pd7) (* (nth 6 other) 0.9))
)

```

```

pz10 (list (+ (car pd11) (* (nth 6 other) 0.6))
          (cadr pd)
        )
pz11 (list (+ (car pd19) 1)
          (- (cadr pd19) (* (nth 6 other) 0.6))
        )
pz12 (list (car pz9)
          (- (cadr pz9) (* (nth 6 other) 0.3))
        )
pz13 (list (- (car pd8) (* (nth 6 other) 0.5))
          (cadr pd8)
        )
pzd1 (polar pd35 (/ (* pi 3) 4) 0.5)
pzd2 (polar pd34 (/ (* pi 5) 4) 0.5)
pz14 (list (- (car pz13) (* (nth 6 other) 0.7))
          (cadr pz13)
        )
pzm1 (list (- (car pz14) (* (nth 6 other) 0.5))
          (+ (cadr pz14) (* (nth 5 other) 0.9))
        )
pzm2 (list (+ (car pd11) (nth 6 other))
          (- (cadr pd11) (nth 5 other))
        )
pa1 (list (+ (car pd11) 0.5)
        (cadr pd11)
      )
)

```

```

(setq cside (ssadd))
(command "zoom" "w" pzm1 pzm2)
(command "layer" "set" layer_base "")
(command "line" pd1 pd2 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd1 pd3 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd2 pd4 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd6 pd5 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd5a1 pd7 "")
(setq cside (ssadd (entlast) cside))
(command "arc" pd6 pd8 pd7)
(setq cside (ssadd (entlast) cside))

```

```

(command "line" pd9 pd6 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd10 pd7 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd12 pd13 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd12 pd14 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd13 pd15 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd14 pd15 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd16 pd18 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd17 pd19 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd18 pd19 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd22 pd23 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd24 pd25 "")
(setq cside (ssadd (entlast) cside))
(command "trim" "o" "q" "" pd3 pd4 pd5 pd5a1 pd8 pd16 pd17 "")
(setq cside (ssadd (entlast) cside))
(command "filletrad" (nth 8 other))
(command "fillet" "trim" "no" pd26 pd27)
(setq cside (ssadd (entlast) cside))
(command "fillet" "trim" "no" pd28 pd29)
(setq cside (ssadd (entlast) cside))
(command "zoom" "w" pzs1 pzs2)
(command "trim" "o" "q" "" pd30 pd31 "")
(setq cside (ssadd (entlast) cside))
(command "zoom" "p")
(command "line" pd37 pd36 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd41 pd40 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd36 pd40 "")
(setq cside (ssadd (entlast) cside))
(command "filletrad" (nth 8 other))
(command "fillet" "trim" "T" pd35 pd38)
(setq cside (ssadd (entlast) cside))
(command "fillet" "trim" "T" pd44 pd34)

```

```

(setq cside (ssadd (entlast) cside))
(command "trim" "o" "q" "" pd9 pd10 pd42 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd43 pd45 "")
(setq cside (ssadd (entlast) cside))
(command "layer" "set" layer_thin "")
(command "line" pd27 pd32 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd28 pd33 "")
(setq cside (ssadd (entlast) cside))
(command "trim" "o" "q" "" pd32 pd33 "")
(setq cside (ssadd (entlast) cside))
(command "layer" "set" layer_sup "")
(command "ltscale" "0.6")
(command "line" pd34 pd35 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd18 pd35 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd19 pd34 "")
(setq cside (ssadd (entlast) cside))
(command "layer" "set" layer_axial "")
(command "line" pd8 pa1 "")
(setq cside (ssadd (entlast) cside))

```

```

(if (= (substr (nth 5 plot_data) 1 1) "y")
  (progn
    (command "layer" "set" layer_dims "")
    (command "dimto" "on")
    (command "dimscale" (* (nth 0 other) 0.04))
    (command "dimlinear" pd43 pd45 pz7)
    (setq cside (ssadd (entlast) cside))
    (command "zoom" "w" pd18 pd27)
    (command "dimradius" pz2 pz3)
    (setq cside (ssadd (entlast) cside))
    (command "zoom" "p")
    (command "zoom" "w" pzb1 pzb2)
    (command "dimradius" pz5 pz6)
    (setq cside (ssadd (entlast) cside))
    (command "zoom" "p")
    (command "dimlinear" pd7 pd34 pz8)
    (setq cside (ssadd (entlast) cside))
    (command "dimlinear" pd8 pd7 pz9)
    (setq cside (ssadd (entlast) cside))
  )
)

```

```

(command "dimlinear" pd15 pd14 pz10)
(setq cside (ssadd (entlast) cside))
(command "dimlinear" pd19 pd13 pz11)
(setq cside (ssadd (entlast) cside))
(command "dimlinear" pd7 pd13 pz12)
(setq cside (ssadd (entlast) cside))
(command "dimlinear" pd7 pd6 pz13)
(setq cside (ssadd (entlast) cside))
(command "dimangular" pzd1 pzd2 pz14)
(setq cside (ssadd (entlast) cside))
)
)

(if (/= (nth 4 plot_data) 0.0)
  (command "rotate" cside "" pt1 (nth 4 plot_data))
)

(defun plot_side (pd / pl1 pl2 pl3)
  (setq pt1 (list (car pd) (cadr pd))
    pd1 (list (car pd)
      (+ (/ (nth 1 other) 2.0) (cadr pd))
    )
    pd1b (list (car pd)
      (- (cadr pd) (/ (nth 1 other) 2.0))
    )
    pd2 (list (+ (/ (nth 5 other) 2.0) (car pd1))
      (cadr pd1)
    )
    pd3 (list (car pd2)
      (- (cadr pd2) (nth 1 other))
    )
    pd4 (list (- (car pd3) (nth 5 other))
      (cadr pd3)
    )
    pd5 (list (car pd4)
      (cadr pd1)
    )
    pd6 (list (+ (car pd1) (/ (nth 5 other) 2.0))
      (cadr pt1)
    )
    pz2 (list (car pd)
      (+ (cadr pd) (/ (nth 5 other) 2.0))

```

```

    )
    pz3 (list (+ (car pd) (nth 1 other))
              (+ (cadr pz2) (nth 1 other))
    )
    pwz1 (list (- (car pd) (/ (nth 5 other) 2.0) (nth 1 other))
              (+ (cadr pd) (/ (nth 5 other) 2.0) (nth 1 other))
    )
    pwz2 (list (+ (car pd) (/ (nth 5 other) 2.0) (nth 1 other))
              (- (cadr pd) (/ (nth 5 other) 2.0) (nth 1 other))
    )
    pwzs1 (list (+ (car pd2) 0.01)
                (+ (cadr pd2) 0.01)
    )
    pwzs2 (list (- (car pd4) 0.01)
                (- (cadr pd4) 0.01)
    )
    pd1c (list (+ (car pd) (/ (nth 5 other) 2.0) (nth 1 other))
              (cadr pd)
    )
    pa1x (list (car pd)
                (+ (cadr pd) (/ (nth 5 other) 2.0) (nth 1 other))
    )
    pa1y (list (car pd)
                (- (cadr pd) (/ (nth 5 other) 2.0) (nth 1 other))
    )
    pa2x (list (+ (car pd) (/ (nth 5 other) 2.0) (nth 1 other))
              (cadr pd)
    )
    pa2y (list (- (car pd) (/ (nth 5 other) 2.0) (nth 1 other))
              (cadr pd)
    )
)

```

```

(setq cside (ssadd))
(command "zoom" "w" pwz1 pwz2)
(command "layer" "set" layer_axial "")
(command "line" pa1x pa1y "")
(setq cside (ssadd (entlast) cside))
(command "line" pa2x pa2y "")
(setq cside (ssadd (entlast) cside))
(command "layer" "set" layer_base "")
(command "circle" pt1 (/ (nth 5 other) 2.0))
(setq cside (ssadd (entlast) cside))

```



```

(command "line" pd4 pd3 "")
(setq cside (ssadd (entlast) cside))
(command "line" pd5 pd2 "")
(setq cside (ssadd (entlast) cside))
(command "zoom" "w" pwzs1 pwzs2)
(command "trim" "o" "q" "" pd4 pd5 pd2 pd3 "")
(setq cside (ssadd (entlast) cside))
(command "zoom" "p")

```

```

(if (= (substr (nth 5 plot_data) 1 1) "y")
  (progn
    (command "layer" "set" layer_dims "")
    (command "dimtoh" "on")
    (command "dimscale" (* (nth 0 other) 0.04))
    (command "dimlinear" pd1 pd1b pd1c)
    (setq cside (ssadd (entlast) cside))
    (command "dimdiameter" pz2 pz3)
    (setq cside (ssadd (entlast) cside))
  )
)

```

```

(if (/= (nth 4 plot_data) 0.0)
  (command "rotate" cside "" pt1 (nth 4 plot_data))
)
)

```

```

(defun DSTU ()
  (setvar "cmdecho" 0)
  (tune_env)
  (setq plot_data (user_input))
  (if (= (substr (nth 2 plot_data) 1 1) "s")
    (plot_side plot_data)
    (plot_main plot_data)
  )
  (setvar "lwdisplay" 1)
  (setvar "cmdecho" 1)
)

```