

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

Другий (магістерський)  
(рівень вищої освіти)

Розроблення методу створення програмних модулів для автоматизованого проектування  
деталей у системі LibreCad  
(тема)

Виконав:

студент 2 курсу, групи КТРСм-22-2

Білошапка І.В.

(прізвище, ініціали)

Спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_

(повна назва освітньої програми)

Керівник Професор Ромашов Ю.В.

(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри КІТАР

(підпис)

Невлюдов І. Ш.

(прізвище, ініціали)

2024р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

11.01.2024

(дата)



(підпис)

Білошапка І.В.

(прізвище, ініціали)

## ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет АКТ  
 Кафедра КІТАР  
 Рівень вищої освіти другий (магістерський)  
 Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології  
 Тип програми Освітньо-професійна  
 Освітня програма Комп'ютеризовані та робототехнічні системи  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_

(підпис)

«» \_\_\_\_\_ 2024р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Білошапка Іван Валерійович  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCad

Затверджена наказом по університету від 03.11.2023 № 1288

2. Термін подання студентом роботи до екзаменаційної комісії 09.01.2024

3. Вихідні дані до роботи LibreCad, переважні розміри, деталі, вал, автоматизоване проектування, дослідження,

4. Перелік питань, що потрібно опрацювати в роботі Аналіз технічного завдання; Вступ; Провести аналіз методів щодо створення програмних модулів автоматизованого проектування деталей; Вибрати систему для розробки автоматизованого проектування; Провести аналіз методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCad; Дослідити автоматизоване проектування деталей з використанням програмних модулів; Провести аналіз впливу вибору ряду переважних розмірів на результат проектування; Провести аналіз впливу стандартів проектування на об'єм проектованої циліндричної деталі; Виробнича безпека в лабораторії; Висновок; Додатки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал представлений у форматі презентації PowerPoint(\*.ppt)10с

6. Консультанти розділів роботи

| Найменування розділу | Консультант<br>(посада, прізвище, ім'я,<br>по батькові) | Позначка консультанта<br>про виконання розділу |      |
|----------------------|---|--|------|
|                      |   | підпис   | дата |
|                      |   |  |      |
|                      |   |  |      |

**КАЛЕНДАРНИЙ ПЛАН**

| №  | Назва етапів роботи   | Термін виконання<br>етапів роботи | Примітка |
|----|---|-----------------------------------|----------|
| 1  | Аналіз технічного завдання  | 30.07.2023                        | виконано |
| 2  | Вступ   | 30.07.2023                        | виконано |
| 3  | Проведення аналіз методів щодо створення програмних модулів автоматизованого проектування деталей | 10.09.2023                        | виконано |
| 4  | Вибір систему для розробки автоматизованого проектування  | 10.11.2023                        | виконано |
| 5  | Дослідження автоматизованого проектування деталей з використанням програмних модулів              | 25.11.2023                        | виконано |
| 6  | Проведення аналізу впливу вибору ряду переважних розмірів на результат проектування               | 20.12.2023                        | виконано |
| 7  | Проведення аналізу впливу стандартів проектування на об'єм проектованої циліндричної деталі       | 29.12.2023                        | виконано |
| 8  | Виробнича безпека в лабораторії   | 04.01.2024                        | виконано |
| 9  | Оформлення пояснювальної записки  | 04.01.2024                        | виконано |
| 11 | Подання роботи на перевірку інтернет-сервісом Unichек   | 10.01.2024                        | виконано |
| 12 | Подання роботи на рецензію  | 14.01.2024                        | виконано |
| 13 | Подання роботи на підпис зав.кафедри  | 14.01.2024                        | виконано |
| 14 | Подання атестаційної роботи в ЕК  | 16.01.2024                        | виконано |

Дата видачі завдання 10.10.2023

Студент \_\_\_\_\_  
(підпис)

Білошапка І.В.  
(прізвище, ініціали)

Керівник роботи \_\_\_\_\_  
(підпис)

Ромашов Ю.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 90 с., 24 рис., 1 дод., 11 джерел.

РОЗРОБЛЕННЯ МЕТОДУ СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ, LibercCad, Python, 2D МОДЕЛЬ, 3D МОДЕЛЬ, АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ДЕТАЛЕЙ.

Об'єкт дослідження – автоматизоване проектування деталей з використанням комп'ютерних систем.

Предмет дослідження – програмні модулі автоматизованих систем проектування деталей.

Мета кваліфікаційної роботи – розроблення та дослідження програмних модулів для автоматизованого проектування деталей у системі LibreCad.

Методи дослідження – метод кінцевих елементів, дисперсійний аналіз, методи ідентифікації, повний факторний експеримент.

У кваліфікаційній роботі досліджено розроблення методу створення програмних модулів для автоматизованого проектування деталей у системі LibercCad.

Для дослідження та розроблення методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD було ознайомлено архітектурою LibreCAD, його API та можливостями. Було вибрана мову програмування яку було використано для розробки модулів. Також було створено основний модуль який буде взаємодіяти з LibreCAD та вибрано завдання для взаємодії для автоматизування їх. Було проведено тестування модуля.

Для забезпечення безпечних умов роботи у лабораторії розробки методу створення програмних модулів для автоматизованого проектування деталей у системі LibercCad проведені необхідні розрахунки з усунення основних шкідливий виробничий факторів в робочій зоні.

Результати кваліфікаційної роботи апробовані у 1 фахової статті та 1 міжнародних конференціях.

## ABSTRACT

Explanatory note: 90 pp., 24 figures, 1 appendix, 11 sources.

DEVELOPMENT OF THE METHOD FOR THE DEVELOPMENT OF SOFTWARE MODULES, LibercCad, Python, 2D MODEL, 3D MODEL, AUTOMATION OF PARTS DESIGN.

The object of consideration is the automation of computer systems and the design of details.

The subject of consideration is software modules of automated detail design systems.

The goal of the qualification work is to develop a method for creating software modules for automated design of parts.

Research methods – children's element methods, dispersion analysis, identification methods, repeated factorial experiment.

In the qualifying work, a method of creating software modules for automated design of details in the LibercCad system was developed.

To study and detail the method of creating software modules for automated design of parts in LibreCAD systems, there is a developed LibreCAD architecture, its API and capabilities. It was my programming that was chosen to develop the modules. A basic module was also created, which will be implemented in LibreCAD and selected for implementation for automation. The module was tested.

To ensure safe robots in the laboratory, methods for creating software modules for automated design of parts in the LibercCad system were developed, raw calculations were carried out using the main fast computing factors in the working area.

The results of the qualifying work were tested at 1 fencing event and 1 international conference.

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ.....  | 8  |
| ВСТУП.....  | 9  |
| 1 АКТУАЛЬНІСТЬ РОБОТИ.....  | 10 |
| 1.1 Аналіз подібних програми LibreCAD .....   | 10 |
| 1.1.1 Аналіз переваг та недоліків AutoCAD .....   | 10 |
| 1.1.2 Аналіз переваг та недоліків Fusion 360 .....  | 13 |
| 1.2 Аналіз недоліків та переваг LibreCAD .....  | 15 |
| 1.3 Розробка методів щодо створення програмних модулів автоматизованого проектування деталей для системи LibreCAD ..... | 19 |
| 2 МЕТОД СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ДЕТАЛЕЙ У СИСТЕМІ LibreCAD.....                  | 22 |
| 2.1 Теоретичні основи автоматизованого проектування деталей .....   | 25 |
| 2.2 Засоби системи LiberCAD для автоматизованого проектування деталей .....   | 30 |
| 2.3 Створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD .....                     | 39 |
| 2.4 Висновок методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD .....     | 47 |
| 3 ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ДЕТАЛЕЙ З ВИКОРИСТАННЯМ ПРОГРАМНИХ МОДУЛІВ .....                            | 49 |
| 3.1 Вплив вибору ряду переважних розмірів на результат проектування .....   | 49 |
| 3.2 Можливості використання типових стандартних рішень щодо проектування деталей.....                                   | 56 |
| 3.3 Вплив стандартів проектування на об'єм проектованої циліндричної деталі .....                                       | 64 |
| 3.4 Виробнича безпека в лабораторії .....   | 67 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....   | 70 |
| ДОДАТОК А Демонстраційний матеріал.....   | 71 |
| ДОДАТОК Б Апробація результатів наукових досліджень.....  | 78 |

## **ПЕРЕЛІК СКОРОЧЕНЬ**

КІТАР – комп’ютерно-інтегрованих технологій, автоматизації та робототехніки;

ХНУРЕ – Харківський національний університет радіоелектроніки;

НДР – науково-дослідна робота;

ПЗ – програмне забезпечення.

## ВСТУП

Сучасне будівництво пристроїв постійно вдосконалюється, і однією зі сфер, які отримують значний науковий і практичний інтерес, є побудова деталей на підприємствах. Ця технологічна розробка має великий потенціал у різних сферах: промислове виробництво, архітектура та будівництво, авіаційна та автомобільна промисловість, машинобудування.

Метою кваліфікаційної роботи є розроблення та дослідження програмних модулів для автоматизованого проектування деталей у системі LibreCad. Для вирішення проблеми потрібно розробити програму яка буде будувати деталь для користувача за його розмірами.

Об'єктом розробки є автоматизоване проектування деталей з використанням комп'ютерних систем.

Предмет розробки є програмні модулі автоматизованих систем проектування деталей.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз програми такі як AutoCAD, LibreCAD, Fusion360;
- провести вибір потрібної програми;
- побудувати блок схему програми;
- вибрати мова програмування;
- провести аналіз теоретичних даних;
- розглянути систему проектування деталі в LiberCAD;
- написати код програми;
- проаналізувати та дослідити вплив переважних розмірів на об'єм, розмір та вплив на деталь;
- оформлення пояснювальної записки керуючись методичними вказівками кваліфікаційної роботи магістра [1] та ДСТУ 3008-15 [2], результати розробки методів щодо створення програмних модулів автоматизованого проектування деталей для системи LibreCAD [3].

## 1 АКТУАЛЬНІСТЬ РОБОТИ

Розробка програмних модулів для автоматизованого проектування деталей у LibreCAD дуже актуальна в сучасних швидкозмінних і конкурентних галузях.

Підвищення ефективності: завдяки автоматизації процесу проектування інженери та дизайнери можуть заощадити час і зменшити кількість помилок у процесі проектування. Це може призвести до збільшення продуктивності, кращої якості дизайну та більш ефективного використання ресурсів.

Економія: автоматизація процесу проектування може призвести до значної економії коштів для підприємств. Зменшуючи час і зусилля, необхідні для проектування деталей, компанії можуть зменшити витрати на проектування та підвищити свою прибутковість.

Вдосконалення виробництва: модулі автоматизованого проектування можуть допомогти знизити ризик помилок і помилок у процесі проектування. Це може призвести до більш якісних конструкцій і продуктів, які з меншою ймовірністю вийдуть з ладу або потребуватимуть дорогої переробки.

Налаштування: модулі автоматизованого проектування можна налаштувати відповідно до конкретних потреб різних галузей промисловості та застосувань. Це може допомогти підприємствам розробити унікальні проекти, які відповідають їхнім конкретним вимогам.

### 1.1 Аналіз подібних програми LibreCAD

Для того щоб обрати систему для якої буде зроблена програма для розробки методу створення програмних модулів для автоматизованого проектування деталей. Буде порівняно низка систем для проектування деталей такі як AutoCAD, LibreCAD та Fusion 360.

#### 1.1.1 Аналіз переваг та недоліків AutoCAD

AutoCAD – це широко поширене програмне забезпечення для автоматизованого проектування (CAD) і креслення, розроблене та продане Autodesk. Це одна з найпопулярніших і всеосяжних доступних програм САПР, яка в основному використовується архітекторами, інженерами, креслярами та іншими професіоналами для створення, редагування та обміну 2D- і 3D-проектуванням і кресленнями.

2D креслення: AutoCAD надає широкий спектр інструментів для створення точних 2D креслень, які можуть включати архітектурні плани, технічні діаграми тощо [4].

3D-моделювання: підтримує створення 3D-моделей, що дозволяє користувачам проектувати та візуалізувати об'єкти та простори в трьох вимірах.

Параметричне проектування: AutoCAD пропонує можливості параметричного проектування, де до об'єктів можна застосовувати зв'язки та обмеження. Це означає, що зміни, внесені до однієї частини проекту, можуть поширюватися на весь креслення.

Великі бібліотеки: AutoCAD постачається з величезною бібліотекою попередньо намальованих об'єктів, символів і компонентів, що може заощадити час і допомогти зберегти узгодженість проектів.

Налаштування: користувачі можуть налаштовувати AutoCAD за допомогою програмування за допомогою AutoLISP або створюючи спеціальні команди та сценарії для автоматизації повторюваних завдань [5].

Співпраця: AutoCAD підтримує спільний доступ до файлів, анотації та функції співпраці, що полегшує роботу кількох користувачів над одним проектом.

Сумісність із різними платформами: AutoCAD доступний для Windows і macOS, що дозволяє користувачам працювати з різними операційними системами.

Інтеграція: його можна інтегрувати з іншими програмними продуктами Autodesk, такими як AutoCAD Civil 3D і AutoCAD Electrical, для виконання спеціалізованих завдань проектування.

AutoCAD використовується в різних галузях, зокрема в архітектурі, інженерії, будівництві, виробництві тощо. Він став стандартом у багатьох професійних робочих процесах проектування та креслення, і його широке застосування пояснюється його потужними функціями та широкою підтримкою та навчальними ресурсами, які надає Autodesk і спільнота користувачів.

AutoCAD і AutoCAD LT підтримують англійську, німецьку, французьку, італійську, іспанську, японську, корейську, китайську упрощену, китайську традиційну, російську, чеську, польську, угорську, бразильську, португальську, данську, голландську, шведську, фінську, норвезьку. Рівень локалізації варіюється від повної адаптації, до перекладу лише довідкової документації.

AutoCAD – це популярне програмне забезпечення для створення двовимірних та тривимірних креслень, що використовується у різних галузях, включаючи архітектуру, машинобудування, електротехніку та інші.

Переваги AutoCAD:

- зручність використання: AutoCAD має простий та інтуїтивно зрозумілий інтерфейс користувача, який дозволяє користувачам легко створювати креслення;
- гнучкість: AutoCAD надає користувачеві можливість редагувати та змінювати креслення у будь-який час, що робить його дуже гнучким та зручним інструментом;
- сумісність: AutoCAD сумісний з більшістю операційних систем та форматів файлів, що дозволяє користувачам працювати з різними програмами та обмінюватися файлами;
- потужні функції AutoCAD має потужні функції, такі як можливість створення тривимірних моделей та анімацій, що робить його ідеальним інструментом для створення складних проектів;
- інтерфейс користувача: AutoCAD має простий і інтуїтивно зрозумілий інтерфейс користувача, що полегшує роботу користувача з програмою (рис. 1.1).

#### Недоліки AutoCAD:

- висока вартість: AutoCAD є одним із найдорожчих програмних продуктів для створення креслень, що може бути проблемою для багатьох користувачів;
- складність: AutoCAD є складним інструментом, який може вимагати тривалого навчання та досвіду, перш ніж користувач зможе використати його на повну потужність;
- високі вимоги до комп'ютера: AutoCAD вимагає потужного комп'ютера з високою продуктивністю, щоб працювати ефективно, що може бути проблемою для користувачів з обмеженими ресурсами;
- не підходить для всіх: AutoCAD може бути не підходящим інструментом для тих, хто не працює з кресленнями та моделями.

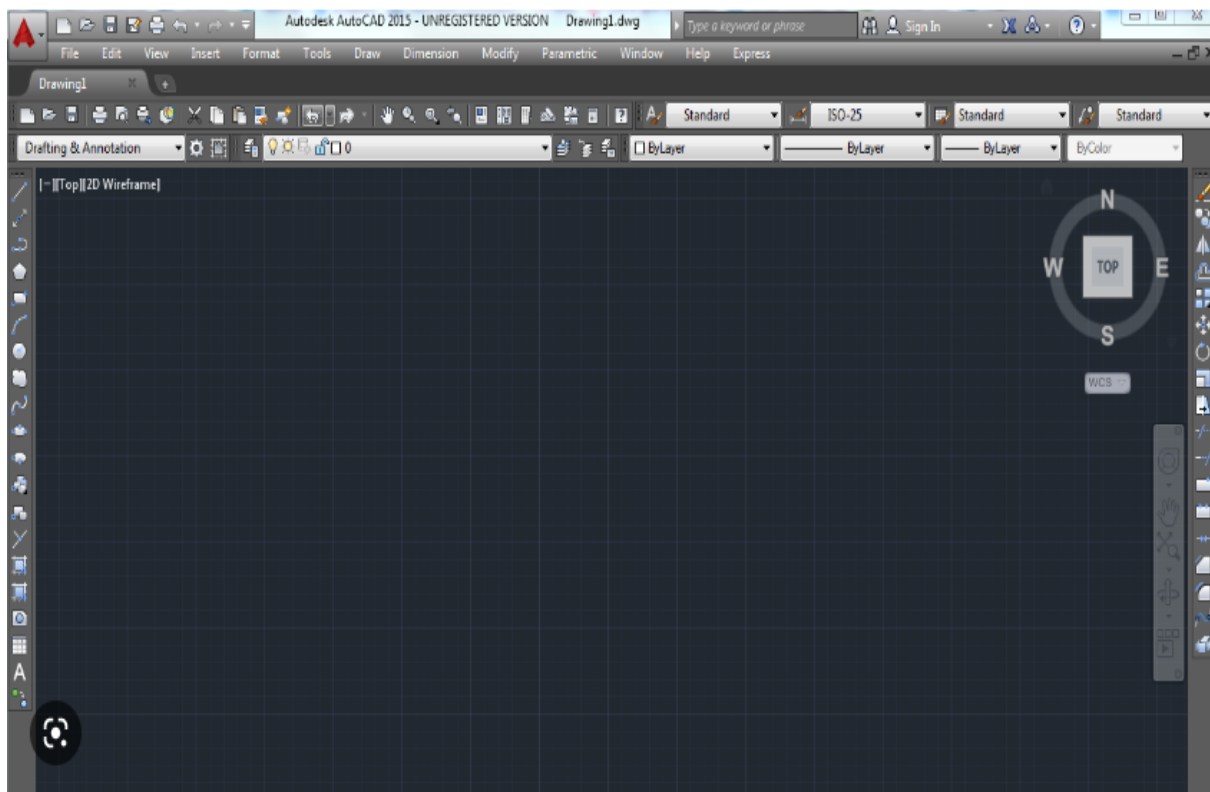


Рисунок 1.1 – Інтерфейс AutoCAD

### 1.1.2 Аналіз переваг та недоліків Fusion 360

Fusion 360 це хмарна програма для 3D-комп'ютерного проектування (CAD), автоматизованого проектування (CAE) і автоматизованого виробництва (CAM), розроблена Autodesk. Fusion 360 призначений для проектування та проектування продуктів, що робить його комплексним інструментом для створення 3D-моделей, імітації конструкцій і створення інструкцій для обробки та 3D-друку [6]. Він використовується в багатьох галузях промисловості, включаючи виробництво, машинобудування та дизайн продукції.

**3D-моделювання:** Fusion 360 надає потужне та інтуїтивно зрозуміле середовище 3D-моделювання для створення складних 3D-проектів і збірок [7].

**Параметричне проектування:** як і AutoCAD, Fusion 360 підтримує параметричне проектування, дозволяючи користувачам створювати зв'язки та обмеження між об'єктами, що полегшує внесення змін у дизайн і збереження цілісності дизайну.

**Моделювання та аналіз:** пропонує інструменти моделювання та аналізу, щоб перевірити, як конструкції працюватимуть за різних умов, наприклад аналіз напруги та температурне моделювання.

**CAM (Computer-Aided Manufacturing):** Fusion 360 включає функції CAM, що дозволяє користувачам генерувати траєкторії для обробки з ЧПУ (комп'ютерне числове

керування) і 3D-друку, що робить його універсальним інструментом для перетворення цифрових конструкцій у фізичні прототипи або продукти.

Співпраця: Fusion 360 базується на хмарі, що означає, що він дозволяє легко співпрацювати між членами команди, які можуть отримувати доступ до одного проекту та редагувати його з різних місць.

Контроль версій: включає функції контролю версій для відстеження ітерацій дизайну та змін з часом.

Візуалізація та анімація: Fusion 360 має вбудовані інструменти візуалізації та анімації для створення реалістичної візуалізації дизайну та анімації продукту.

Сумісність із різними платформами: Fusion 360 доступний як для Windows, так і для macOS, і до нього можна отримати доступ через веб-браузери, що робить його універсальним інструментом для користувачів на різних платформах.

Fusion 360 користується особливою популярністю серед малих і середніх підприємств і індивідуальних дизайнерів завдяки повному набору функцій і тому факту, що він доступний як послуга на основі передплати. Хмарний характер Fusion 360 також означає, що його можна використовувати з різних пристроїв, а дані про проектування можна легко обмінювати та отримувати доступ із багатьох місць.

Fusion 360 – це інтегроване програмне забезпечення для проектування, моделювання та виробництва, яке використовується у різних галузях, включаючи машинобудування, електротехніку та архітектуру.

Переваги Fusion 360:

- інтеграція: Fusion 360 поєднує в собі функціональність декількох інструментів, таких як CAD, CAM та CAE, що робить його ідеальним інструментом для конструювання, аналізу та виробництва;

- хмарна платформа: Fusion 360 працює на хмарній платформі, що полегшує доступність та обмін файлами між користувачами та пристроями;

- інтерфейс користувача: Fusion 360 має простий і інтуїтивно зрозумілий інтерфейс користувача, що полегшує роботу користувача з програмою (рис. 1.2);

- потужні функції Fusion 360 має безліч потужних функцій, таких як можливість створення тривимірних моделей, аналізу міцності та виробничої підготовки, що робить його ідеальним інструментом для створення складних проектів.

Недоліки Fusion 360:

- висока вартість Fusion 360 є платним програмним забезпеченням, що може бути проблемою для багатьох користувачів;

- обмежена підтримка файлів Fusion 360 має обмежену підтримку файлів, що може бути проблемою для користувачів, що працюють з різними форматами файлів;
- необхідність підключення до Інтернету: Fusion 360 потребує підключення до Інтернету для роботи, що може бути проблемою для користувачів у регіонах з обмеженим доступом до Інтернету;
- складність: Fusion 360 є складним інструментом, який може вимагати тривалого навчання та досвіду, перш ніж користувач зможе використати його на повну потужність.

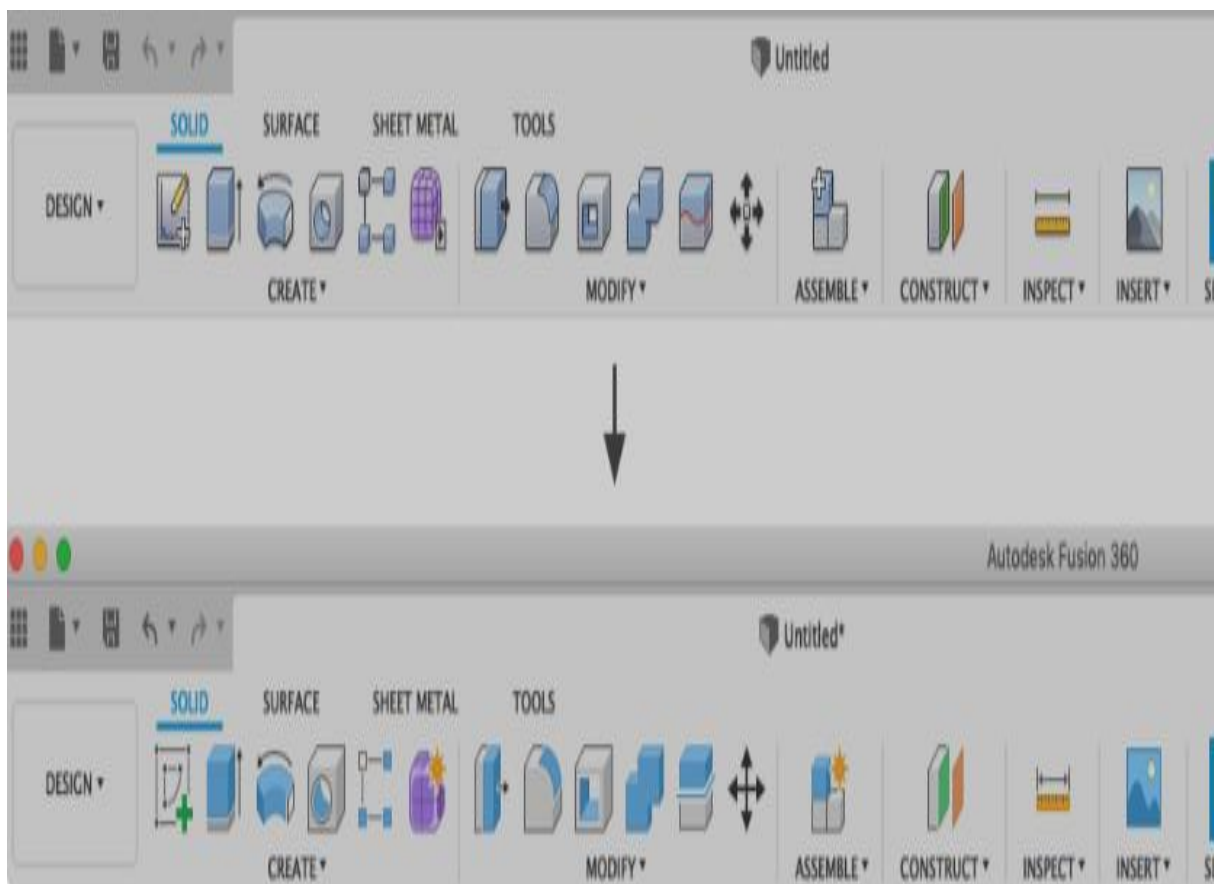


Рисунок 1.2 – Інтерфейс Fusion 360

## 1.2 Аналіз недоліків та переваг LibreCAD

LibreCAD – це програма двовимірного автоматизованого проектування (CAD) з відкритим кодом. Це безкоштовний проект, керований спільнотою, який надає користувачам інструменти для створення двовимірних технічних креслень і дизайнів. LibreCAD подібний за функціональністю до іншого популярного програмного забезпечення 2D CAD, наприклад AutoCAD.

Інструменти малювання: пропонує різноманітні інструменти малювання, включаючи лінії, кола, дуги, еліпси та текст, які необхідні для створення технічних креслень.

Шари: користувачі можуть організовувати свої малюнки в шари, що полегшує керування складними проектами та керування видимістю різних елементів.

Вимірювання та визначення розмірів: LibreCAD надає інструменти для додавання точних вимірювань і розмірів до ваших креслень, що важливо для інженерних та архітектурних застосувань [8].

Прив'язка та сітка: функції прив'язки та сітки допомагають користувачам точно вирівнювати елементи на своїх малюнках.

Імпорт і експорт: LibreCAD підтримує поширені формати файлів САПР, такі як DXF і DWG, що робить його сумісним з іншим програмним забезпеченням САПР. Ви також можете експортувати свої проекти в різні формати зображень.

Розширюваність: дозволяє створювати плагіни та розширення, що дозволяє користувачам налаштовувати програмне забезпечення відповідно до своїх конкретних потреб.

LibreCAD особливо популярний серед користувачів, яким потрібні можливості 2D-креслення, але вони віддають перевагу відкритим кодам і безкоштовним програмним рішенням. Він доступний для різних операційних систем, включаючи Windows, macOS і Linux.

LibreCAD (раніше CADuntu) – кросс-платформова та вільна САПР для 2D креслення та проектування, створена на основі QCAD. LibreCAD дозволяє виконувати завдання двомірного проектування, такі як підготовка інженерних та будівельних креслень, схем та планів [9]. Працює під управлінням операційних систем Linux, Windows та Mac OS. Розповсюджується під ліцензією GPLv2. Початковий код написаний на мові програмування C++ з використанням бібліотек Qt.

#### Переваги та недоліки LibreCAD

– безкоштовне програмне забезпечення з відкритим вихідним кодом: LibreCAD – це безкоштовне програмне забезпечення з відкритим вихідним кодом, що означає, що воно доступне будь-кому для завантаження, використання та зміни. Це робить його доступною та доступною альтернативою комерційним програмним засобам САПР;

– сумісність із різними платформами: LibreCAD працює в кількох операційних системах, включаючи Windows, macOS і Linux, що робить його доступним для широкого кола користувачів;

– зручний інтерфейс LibreCAD має зручний інтерфейс, у якому легко орієнтуватися навіть для тих користувачів, які тільки починають працювати з програмним забезпеченням САПР (рис. 1.3);

– широкий набір інструментів: LibreCAD надає повний набір інструментів двовимірного проектування, включаючи геометричні форми, розміри та текстові анотації, які зазвичай використовуються в різних галузях промисловості;

– підтримка DWG і DXF: LibreCAD підтримує популярні формати файлів DWG і DXF, що означає, що користувачі можуть відкривати та редагувати файли, створені в інших програмних інструментах САПР;

– налаштування: LibreCAD дуже легко налаштовується, з різними плагінами та доповненнями доступні для розширення його функціональності та задоволення конкретних потреб користувачів;

– активна спільнота: LibreCAD має активну спільноту розробників і користувачів, які надають підтримку, роблять внесок у її розвиток і діляться своїми знаннями та досвідом;

– та основною перевагою є можливість проектувати деталь завдяки блокноту або іншого текстового редактору.

#### Недоліки

– обмежені можливості: LibreCAD має менше функціональних можливостей, порівняно з платними аналогами, такими як AutoCAD або SolidWorks. Це може бути проблемою для більш складних проектів, де потрібна більша гнучкість;

– відсутність підтримки 3D-моделювання: LibreCAD призначений для 2D-моделювання, що робить його непридатним для проектування 3D-об'єктів;

– система підтримки: LibreCAD має обмежену кількість користувачів та розвивається менш швидко, ніж комерційні програми, що може призвести до відсутності підтримки, оновлень і виправлень помилок;

– недостатня документація: LibreCAD має обмежену документацію та навчальні матеріали, що може ускладнити навчання програмі.

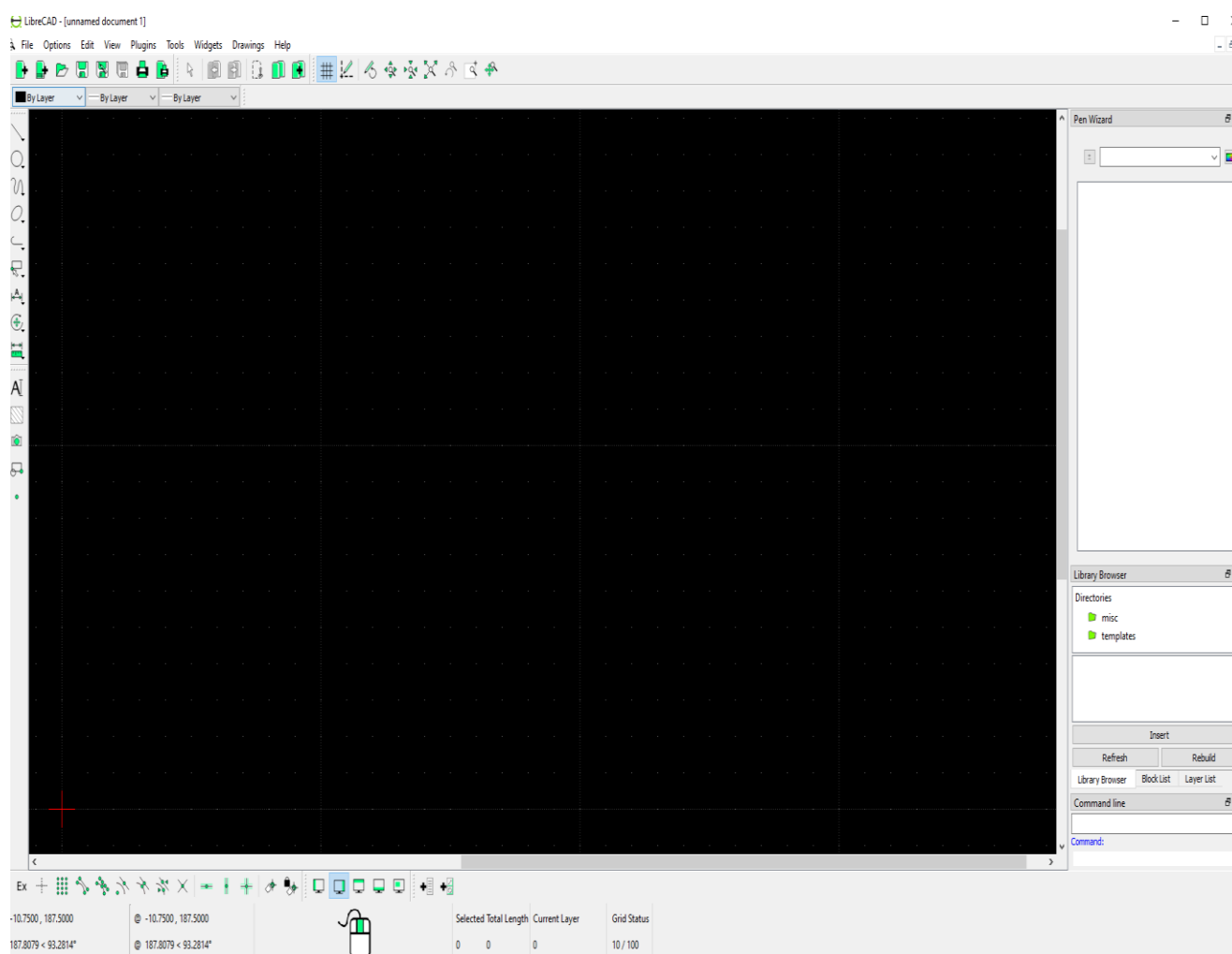


Рисунок 1.3 – Інтерфейс LibreCAD

Всі ці недоліки можуть стати проблемою для користувачів LibreCAD, особливо якщо вони працюють над складними проектами, що потребують багатофункціональної програми з підтримкою 3D-моделювання.

Однак при належному використанні LibreCAD може бути дуже корисним інструментом для проектування 2D-об'єктів.

В ході вибору та аналізу програм для модулювання було вибрано LibreCAD так як він безкоштовний, він підтримує всі операційні системи, зручний для використання, має широкий набір інструментів, підтримує популярні формати файлів, дуже легко налаштовується з різними плагінами, дуже важлива перевага можливість проектувати деталь завдяки блокноту або іншого текстового редактору.

AutoCAD, як вже зазначалося, є гігантом у галузі черчення. Можна сказати, що це Rolls Royce серед CAD із відповідним ціновим брендом. DraftSight можливо є №2 у цій галузі; обмежена версія є безкоштовною для особистого використання, але для корпоративних користувачів ціна залежить від розміру компанії. Однак я не знайшов

прямого конкурента для LibreCAD. Основні характеристики, які я розглядаю, це 2D, вільне та відкрите програмне забезпечення, без обмежень (немає обмеженої версії, без випробувального періоду, без необхідності реєстрації, без встановленого додаткового програмного забезпечення тощо), досить розвинене рішення, а також питання форматів файлів (зокрема .dxf та форматів, придатних для друку, з PDF на першому місці).

Формат .dwg (для рисунків) є важливим для обміну малюнками між CAD-програмами. Зараз це є експериментально в LibreCAD, але, як повідомляється, відповідає документам, збереженим в AutoCAD 2007 та попередніх версіях. Ви можете знайти невеликий тест DWG, який я провів, зовсім в кінці цього посібника.

Відзначте безкоштовне програмне забезпечення LibreOffice Draw. На сьогодні в нього відсутні важливі функції, необхідні для технічного малювання (наприклад, формат файлу DXF), але з розширеннями воно може стати легким альтернативним рішенням для CAD – час покаже. Draw вже має 3D-можливості, перевершуючи LibreCAD.

### 1.3 Розробка методів щодо створення програмних модулів автоматизованого проектування деталей для системи LibreCAD

Загалом розробка програмних модулів для автоматизованого проектування в LibreCAD вимагає поєднання технічних навичок, знань предметної області та уваги до деталей. За допомогою правильної методології та процесу розробки ці модулі можуть допомогти оптимізувати процес проектування та підвищити продуктивність для користувачів у різноманітних галузях промисловості. Використовуючи потужність автоматизації, інженери та дизайнери можуть покращити свою продуктивність, зменшити витрати та розробити високу якість конструкції, які відповідають їхнім конкретним потребам.

В ході роботи було розглянуто та проаналізовані системи які підходили для модулювання деталей того ж типу що і LibreCAD. Було проаналізовано переваги та недоліки таких програм та прийнято рішення вибрати LibreCAD так як він безкоштовний та має можливість створювати деталі за допомогою текстового формату.

Для розробки методів створення програмних модулів автоматизованого проектування деталей для системи LibreCad, для початку буде створено блок схему (рис. 1.4). На цій блок схемі зображено хід роботи програми. Після того як програма почалась ми введемо дані для розрахунку та креслення деталей.

Далі в програмі проходить ініціалізація данні стандарту відповідно обраному стандарту. Після чого програма проводить розрахунок розмірів деталі з умов міцності. Далі програма за даними які було введено створює файл формат текстового документа після чого він генерується на жорсткому диску.

Після завершення програми ми можемо завантажити створений файл в систему LibreCAD де вже буде накреслена потрібна деталь.

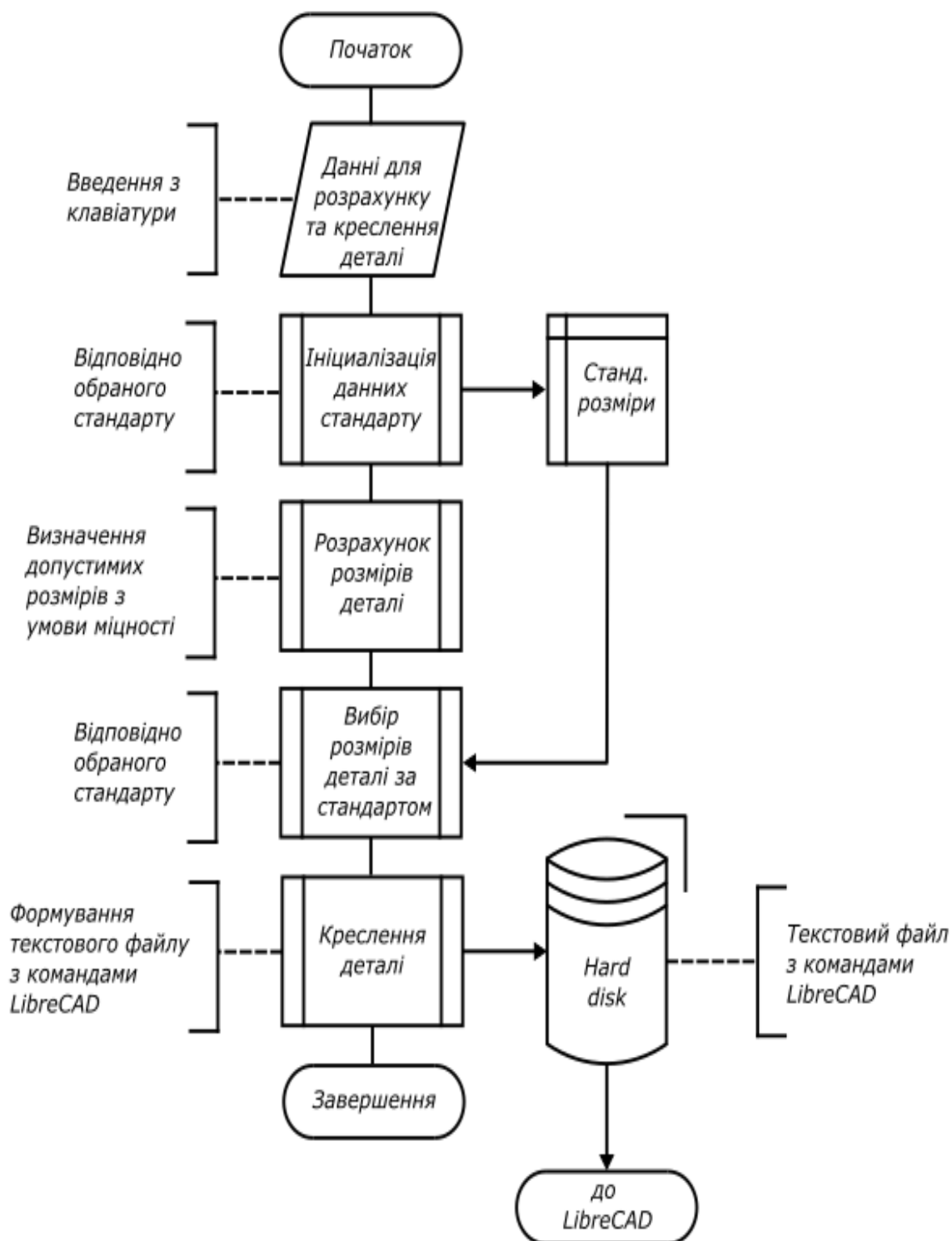


Рисунок 1.4 – Блок схема

Було розроблено блок схему яка описує весь процес роботи програми та взаємодію с LibreCAD. Для роботи с програмою потрібно мати дані для виробництва деталей такі як розмір, товщина, довжина та ширина. Програма буде створювати файл формату текстового документа який можна завантажити в LibreCAD. AutoCAD, як вже зазначалося, є гігантом у галузі черчення. Можна сказати, що це Rolls Royce серед CAD із відповідним ціновим брендом.

Також було досліджено предмет та об'єкт практики та було досягнуто цілей такі як:

- проведено аналіз програми такі як AutoCAD, LibreCAD та Fusion360;
- було вибрано потрібної програми;
- було побудовано блок схему програми.

## 2 МЕТОД СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ДЕТАЛЕЙ У СИСТЕМІ LibreCAD

Для початку треба вибрати середу розробки програми. На вибір було підібрано декілька мов програмування переглянемо всі переваги які нам знадобляться для розробки програми. Переглянемо всі із варіантів:

– спочатку розглянемо Python.

Python – це високорівнева, інтерпретована мова програмування загального призначення, яка збільшила доступність завдяки своїй простоті, читабельності коду та широкому спектру програм.

Простий синтаксис: Python відомий своїм лаконічним та легким для читання синтаксисом. Це дозволяє розробникам швидко написати код і допомогти його розумінню.

Інтерпретованість: Python – це інтерпретована мова, що означає наявність необхідності в явній компіляції перед виконанням. Це полегшує тестування та розробку прототипів.

Кросплатформенність: код, написаний на Python, може працювати на різних операційних системах без змін, що робить його кросплатформеним.

Широкий спектр бібліотек: Python має велику кількість бібліотек для різних областей, таких як наукові обчислення (NumPy, SciPy), робота з базами даних (SQLAlchemy), веб-розробка (Django, Flask), машинне навчання (TensorFlow, PyTorch) та багато інших.

Спільнота: Python має велику та активну спільноту розробників, яка підтримує відкритий код та надає безліч ресурсів для навчання та підтримки.

Висока продуктивність: хоча Python може бути повільнішим для деяких інших мов програмування, таких як C++ або Java, він все буде мати високу продуктивність завдяки оптимізації з використанням пам'яті та швидкої розробки.

Python використовується для широкого спектру завдань, включаючи веб-розробку, аналіз даних, наукові дослідження, штучний інтелект, автоматизацію, ігрову розробку та багато іншого;

– розглянемо C++.

C++ – це потужна та високопродуктивна об'єктно-орієнтована мова програмування, яка поєднує в собі можливості мови C з об'єктно-орієнтованим програмуванням.

Швидкодія: C++ дозволяє ефективно використовувати ресурси системи, забезпечуючи високу швидкість виконання програми.

Об'єктно-орієнтоване програмування: мова підтримує концепцію об'єктно-орієнтованого програмування, такого як класи та успадкування, що полегшує створення структурованих та повторно використовуваних кодових баз.

Низькорівневий доступ до пам'яті: C++ надає розробникам контроль над використанням пам'яті, що є корисним для оптимізації та роботи з апаратним забезпеченням.

Широкі можливості: відповідно до стандарту C++, мова має багатий набір бібліотек і фреймворків, що дозволяє розробникам ефективно використовувати готові рішення для різних завдань.

Підтримка багатозадачності: C++ надає можливості для розробки багатопотокових програм, що важливо для роботи в сучасних багатозадачних середовищах.

Широке використання: використовується для розробки високоефективних програм, таких як операційні системи, вбудовані системи, графічні додатки, ігри та інші.

C++ володіє великою кількістю прикладів використання в промисловості та є популярним вибором для проектів, де важлива продуктивність та контроль над ресурсами;

– розглянемо Java.

Java – це об'єктно-орієнтована мова програмування, яка знає вашу кросплатформенність та високий рівень портативності. Основні характеристики Java включають:

– кросплатформенність: Java-код може працювати на різних платформах без необхідності перекомпіляції, застосовуючи віртуальну машину Java (JVM);

– об'єктно-орієнтоване програмування: мова сприяє використанню об'єктно-орієнтованих концепцій, таких як класи та успадкування;

– багатопотоковість: Java надає вбудовану підтримку багатопотоковості, що дозволяє розробникам легко створювати багатозадачні програми;

– безпека: мова використовує механізми безпеки, такі як віртуальна машина та механізми контролю доступу, для створення безпечних додатків;

– широкий екосистема: Java має велику кількість бібліотек, фреймворків та інструментів, що полегшує розробку різноманітних додатків;

– висока продуктивність: хоча Java не така продуктивна, як мови низькорівневого рівня, вона дає достатню продуктивність для багатьох видів застосування;

– популярність у корпоративному середовищі: Java широко використовується для розробки корпоративних додатків, веб-систем, мобільних додатків на Android, сервлетів та інших великих проектів.

Java залишається однією з найпопулярніших програм програмування через свою кросплатформенність, велику спільноту розробників та високий рівень надійності.

– розглянемо C#.

C# (C-Sharp) – об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Основні характеристики C# включають.

Інтеграція з платформою .NET: C# є ключовою мовою для розробки на платформі .NET, що включає в себе велику кількість бібліотек і фреймворків для різних програм.

Об'єктно-орієнтоване програмування: мова підтримує об'єктно-орієнтовану концепцію, таку як класи та успадкування.

Спрощений синтаксис: C# володіє читабельним та лаконічним синтаксисом, що полегшує розробку.

Автоматичне управління пам'яттю: C# використовує збиральник вмісту для автоматичного вивільнення невикористаних ресурсів.

Безпека: мова надає механізми безпеки, включаючи контроль типів та обмеження доступу до ресурсів.

Багатопотоковість: C# підтримує багатозадачність та асинхронне програмування для ефективної роботи з багатьма операціями одночасно.

Широке використання в розробці веб-додатків та мобільних додатків: використання C# для розробки веб-додатків ASP.NET, десктопних додатків, а також мобільних додатків на платформі Xamarin.

Платформонезалежність через .NET Core: Введення .NET Core дозволяє писати кросплатформенний додаток на C# для Windows, Linux та macOS.

C# володіє активною спільнотою розробників і використовує в різноманітних сферах розробки, включаючи корпоративні, веб-та мобільні додатки.

Із перерахованих мов більше за все підходить Python щодо вибору мови програмування для розробки модулів для LibreCAD, можна порівняти Python з іншими мовами програмування наступним чином:

а) Python проти C++;

1) Python: забезпечує високий рівень абстракції, що полегшує розробку. Проте для завдань, де важлива продуктивність та низький рівень ресурсомісткості, Python може бути менш ефективним;

2) C++: має високу продуктивність та широкі можливості для роботи з пам'яттю, що важливо для оптимізації продуктивності CAD-системи. Використовується у багатьох великих проєктах своєю швидкістю;

б) Python проти Java;

1) Python: має лаконічний синтаксис, призначений для зручності розробників та читачів коду. Ідеально підходить для швидкого розвитку та прототипування завдяки високій рівню абстракції;

2) Java: має велику кросплатформенність і високий рівень абстракції. Зазвичай використовується для побудови кросплатформених додатків, проте може мати більші витрати пам'яті та меншу продуктивність разом із C++;

в) Python проти C#;

1) Python: має велика кількість бібліотек для різних завдань, що дозволяє розробникам швидко реалізовувати функціонал. Має активну спільноту розробників та багато ресурсів для навчання;

2) C#: широко використовується в розробці програм для платформи .NET, має високий рівень абстракції та зручний синтаксис, але менш підходить для системного програмування.

Після порівняння мов було остаточно вирішено використовувати Python в ролі мови для створення програмних модулів для автоматизованого проєктування деталей у системі LibreCAD.

## 2.1 Теоретичні основи автоматизованого проєктування деталей

Теоретична основа вказує на систему теоретичних знань, принципів та концепцій, на яких базується деяка наукова чи науково-технічна галузь. Теоретична основа служить основою для побудови концепції, розробки методів та вибору інструментів для вирішення конкретних завдань чи вирішення проблем у визначеній області. Вона дозволяє науковцям, інженерам чи проєктувальникам розуміти основи та принципи своєї роботи, а також розробляти нові ідеї та методи на основі цієї теоретичної бази.

Теоретичні аспекти автоматизованого проектування деталей базуються на ряді основних принципів та зрозуміти, які визначають ефективність та точність процесу створення технічних рішень.

Геометричне моделювання – є ключовою частиною теоретичних аспектів автоматизованого проектування. Воно використовується для представлення форм та розмірів деталей у віртуальному просторі. Математичні поняття, такі як точки, лінії, поверхні та об'єми, використовуються для створення геометричних об'єктів.

Параметричне моделювання – визначається параметрами, які визначають форму та розміри деталі. Зміна цих параметрів може автоматично змінювати геометричну модель, що дозволяє швидко вносити зміни та експериментувати з різними варіантами проекту.

Використання математичних моделей – є необхідним для опису форм та взаємодії між різними елементами конструкції. Диференціальні рівняння, матричні перетворення та алгебраїчні вирази дозволяють аналізувати та оптимізувати геометричні параметри деталей.

Алгоритми генерації дозволяють автоматично створювати геометричні об'єкти на основі вхідних параметрів, тим самим прискорюючи процес проектування. Алгоритми оптимізації використовуються для автоматичного визначення оптимальних значень параметрів з урахуванням заданих обмежень та критеріїв якості.

Комп'ютерне моделювання та імітація дозволяють вирішувати складні інженерні завдання перед фізичним виготовленням прототипу. Це дозволяє виявити та усунути проблеми ще на етапі віртуального проектування, що економити час та ресурси.

Деталь – це геометричний об'єкт який являється частиною більшої конструкції або системи. Деталі є конкретними частинами чи елементами, які складають цілісну конструкцію чи механізм. Вони можуть бути виготовлені з різних матеріалів, таких як метал, пластик, дерево, іноді скло чи композитні матеріали, залежно від призначення та вимог конкретного додатку.

У кожній деталі є свої кращі розміри вони підбираються за такими різними параметрами як:

- функціональність: розміри деталей повинні відповідати вимогам та функціональності системи, до якої вони входять. Наприклад, в механічних системах розміри деталей можуть бути оптимізовані для забезпечення ефективного виконання робочих операцій;

– міцність та вага: у металургії та інших галузях, де міцність є ключовим фактором, розміри деталей можуть бути вибрані таким чином, щоб забезпечити необхідну міцність та стійкість при мінімальній масі;

– ергономіка: дизайні виробів для споживачів важливо враховувати ергономічні аспекти, такі як зручність та безпека використання;

– стандартизація: розміри деталей можуть бути вибрані відповідно до існуючих стандартів або норм, що полегшує виробництво та обслуговування;

– естетика: у деяких випадках, таких як дизайн виробів для масового ринку, розміри деталей можуть бути обрані з огляду на естетичні аспекти та вимоги ринку.

Важливо враховувати всі ці перелічені чинники та забезпечити баланс між ними залежно від конкретного завдання чи проекту. Кращі розміри деталей визначаються в контексті конкретних вимог та обставин, і вони можуть різнитися від випадку до випадку.

Щоб побудувати деталь потрібно кращі числа їх так називають тому, що їм слід віддавати перевагу порівняно з іншими числами при встановленні значень параметрів та розмірів машинобудівної продукції.

Значення головних та основних параметрів виробів утворюють ряди (параметричні ряди), побудовані за певною математичною закономірністю. Параметричні ряди утворюють, наприклад, розміри взуття та одягу, діаметри шарикопідшипників, вантажопідйомність автомашин, напруги електричної мережі, потужності електричних машин.

Для розробки параметричних рядів знадобилося насамперед встановлення єдиної закономірності в системі уніфікованих та стандартизованих величин, до яких відносять не тільки геометричні розміри, але й такі параметри, як потужність, вантажопідйомність, продуктивність, швидкість і т.д. Встановлення єдиної закономірності таких величин мало забезпечити ув'язування номінальних характеристик окремих видів виробів між собою та забезпечити їх наступність по всьому технологічному циклу виробництва та споживання. Такою системою є обґрунтований ряд чисел, застосування яких забезпечує узгодження параметрів та розмірів окремого виробу з усіма пов'язаними з ним видами продукції (наприклад, розміри виробів та розміри тари та упаковки).

Кращі числа отримані на основі геометричної прогресії формула (2.1).

$$g_i = \pm 10^{\frac{i}{R}}, \quad (2.1)$$

де  $R = 5, 10, 20, 40, 80, 160$ ;

$i$  – ціле число від 0 до  $R$ .

Значення  $R$  визначає кількість членів прогресії в одному десятковому інтервалі. Переважні числа є округлені значення членів нижчої даної прогресії.

Члени прогресії, які розташовані у інтервалі від 1.00 до 10.00, становлять вихідний ряд. Розрізняють такі ряди кращих чисел: основні ряди кращих чисел; додаткові ряди переважних чисел; складові ряди переважних чисел; наближені кращі числа; похідні кращі ряди чисел; спеціальні ряди чисел.

Переважні числа та їх ряди використовують: при встановленні стандартних значень та рядів стандартних значень величин; при нормуванні значень вихідних параметрів продукції, умов її існування та процесів, а також дозволених та допустимих відхилень; у разі нормування значень параметрів продукції, пов'язаних з логарифмічною залежністю з вихідними параметрами, значення яких нормуються за допомогою переважних чисел; при наведенні значень параметрів предметів та процесів (у тому числі природних констант), якщо використання кращих чисел не тягне за собою виходу за межі допустимого відхилення.

Похідні та спеціальні ряди чисел допускається застосовувати лише у разі, якщо застосування рядів кращих чисел (основних, додаткових, складових) неможливе або недоцільне. У разі альтернативних варіантів перевагу слід віддавати ряду, що має менше градацій, а також основному ряду перед вибором.

Похідні та спеціальні ряди чисел допускається застосовувати лише у разі, якщо застосування рядів кращих чисел (основних, додаткових, складових) неможливе або недоцільне. У разі альтернативних варіантів перевагу слід віддавати ряду, що має менше градацій, а також основному ряду перед вибором роковим та складовим.

Застосування додаткових рядів переважних чисел та переважних рядів чисел допускається лише в тому випадку, якщо ряд  $R40$  або створень на його основі похідний ряд чисел не забезпечує необхідної кількості градацій.

Не допускається утворювати складові ряди шляхом з'єднання переважних рядів різних видів, наприклад, геометричного та арифметичного, комплементарного та геометричного тощо.

Ще одна із формул яку потрібно розглянути це напруження у статично навантажених деталях формула (2.2).

$$\sigma = \frac{F}{A}, \quad (2.2)$$

де  $\sigma$  – це напруження (у паскалях або  $\text{Н/м}^2$ );

$F$  – сила яка діє на деталь (у Ньютонах);

$A$  – площа перерізу деталей (в метрах квадратних).

Ця формула використовується для розрахунку внутрішніх сил та деформацій у матеріалі деталі. Ця формула допомагає визначити, наскільки сильно матеріал деталі деформується або піддається напруженню під дією зовнішніх сил. У статичних умовах (якщо сила не змінюється з часом), це напруження може бути визначено для точок усередині матеріалу, допомагаючи інженерам оцінити стійкість та безпеку конструкції.

Момент інерції ( $I$ ) визначається для геометричних об'єктів та вказує на їх ступінь зосередження маси щодо осі обертання. Для прямокутного перерізу, такого як прямокутна пластина або брус, момент інерції може бути обчислений за допомогою наступної формули (2.3)

$$I = \frac{B \cdot h^3}{12}, \quad (2.3)$$

де  $I$  – момент інерції відносно осі, яку ви розглядаєте (в квадратних метрах чи метрах до четвертого ступеня);

$b$  – ширина прямокутного перерізу (в метрах);

$h$  – висота прямокутного перерізу (в метрах).

Ця формула застосовується до прямокутного перерізу, коли обертання відбувається вздовж осі, що перпендикулярна до широкої сторони прямокутника (осі, яка проходить через центр маси паралельна короткій стороні). Формула виведена з використанням основних принципів теорії моменту інерції та геометричних властивостей прямокутного перерізу.

Ще одна формула яка може знадобитися закон Гука для пружин вона визначає залежність між силою, яка діє на пружину, та її деформацією за формулою (2.4).

$$F = k \cdot \delta, \quad (2.4)$$

де  $F$  – сила, що діє на пружину (у ньютонах);

$k$  – константа пружності пружини (у ньютонах на метр);

$\delta$  – величина відхилення (деформації) пружини від її початкового стану (у метрах).

Ця формула стверджує, що сила, яка діє на пружину, пропорційна величині її деформації. Константа пружності ( $k$ ) визначає жорсткість пружини та вказує, наскільки

важко її деформувати. Чим більше значення  $k$ , тим жорсткіша пружина. Цей закон часто використовується для опису деформацій та поведінки пружин у фізиці, інженерії та інших науках. Закон Гука є дуже ефективним при аналізі пружин в умовах малих деформацій, коли їхнє відхилення від початкового стану не є надто великим.

Виконавши аналіз теоретичних основ автоматизованого проектування деталей було розглянуто декілька формул які можуть знадобитися в майбутньому також розглянуто декілька термінів та які різновидності деталей існує.

## 2.2 Засоби системи LiberCAD для автоматизованого проектування деталей

Термін сутність відноситься до графіки об'єкт («віджет») у системі САПР. Типові сутності, які підтримують більшість CAD системи, це:

- точка;
- лінії;
- прямокутники;
- дуги, кола та еліпса.

Більш складні сутності включають по лінії, тексти, розміри, штрихування (близько розташовані паралельні прямі) і Spline (криві, що з'єднують два або більш конкретні моменти) (рис. 2.1).

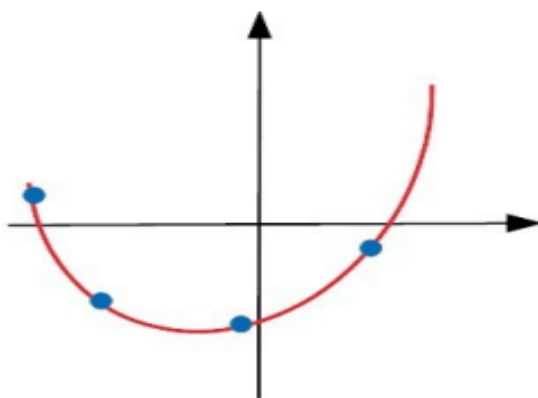


Рисунок 2.1 – Spline

Одним з основних концептів в комп'ютерному проектуванні за допомогою CAD є використання шарів для організації малюнка. Кожен об'єкт в малюнку розташований на одному шарі, і на одному шарі може бути безліч об'єктів.

Зазвичай об'єкти із спільною "функцією" чи спільними властивостями розміщуються на одному шарі. Наприклад, ви можете помістити всі осі на кулю під назвою "axes" (осі).

Кулі можуть мати свої власні властивості (колір, ширина лінії, стиль лінії тощо). Кожний об'єкт може мати свої власні властивості або властивості можуть бути визначені кулею, на якій він розташований. У разі останнього ви можете змінити, наприклад, колір усіх об'єктів на шарі, встановивши колір (червоний, наприклад) для цієї кулі [10].

Блок – це група об'єктів, схожа на те, що ви створюєте за допомогою функцій "Група" та "Розгрупувати" в LibreOffice, Impress і PowerPoint. Блоки можуть бути вставлені в той самий малюнок більше одного разу з різними атрибутами, в різних місцях та з різними масштабами та кутами обертання. Такий блок зазвичай називається вставкою.

Вставки мають атрибути, так само як об'єкти та кулі. Об'єкт, який є частиною вставки, може мати свої власні атрибути або частину атрибутів вставки.

Після створення вставок вони все ще пов'язані з блоком, який вони представляють. Сила вставок полягає в тому, що ви можете змінити блок один раз, і всі вставки будуть відповідно оновлені (аналогічно використанню стилів у обробці тексту).

Для оптимального використання LibreCAD розуміння системи координат та принципів роботи координат є важливим. Зазвичай використовують два основні типи координат: декартові та полярні (рис. 2.2).

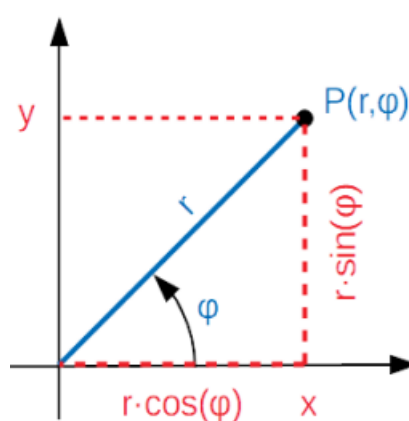


Рисунок 2.2 – Декартова (x,y) і полярна (r,φ) системи координат

Декартова система координат – це загальноприйнята система, що використовується в програмах САД. Конкретна точка на малюнку визначається точними відстанями від осей x та y. Наприклад, точка в малюнку може бути визначена як 60,45 ([x,y], слід звернути увагу на кому, яка розділяє обидва числа).

Система полярних координат використовує одну відстань і один кут для визначення точки на малюнку. Наприклад, точка на малюнку може бути визначена як  $50<45$ , що означає відстань 50 одиниць (r) і кут 45 градусів ( $\phi$ , слід звернути увагу на знак  $<$ , який використовується для кута).

З абсолютними координатами координатні точки вводяться у прямому відношенні до початку координат 0,0. Для цього в LibreCAD введіть точку у відповідних одиницях, наприклад, 60,45.

У відносних координатах координатні точки вводяться відносно попередньо введеної точки (не щодо початку координат). Наприклад, якщо ваша перша точка – 20,45, ви потім вводите наступну точку щодо цієї, використовуючи символ @. Таким чином, @50,50 розташує другу точку 50 одиниць горизонтально вздовж вісі X та 50 одиниць вертикально вздовж осі Y, отримуючи координату 70,95.

Розміри об'єктів у малюнку передаються за допомогою вимірювання. Виміри відстаней можуть бути показані двома стандартизованими формами: лінійні та систему ординат.

Лінійні виміри – це звичайний випадок, де використовується дві паралельні лінії, названі лініями виносу, розташовані на відстані між двома заданими точками. Лінія, перпендикулярна цим лініям виносу, називається лінією вимірювання, зі стрілками на її кінцях (рис. 2.3). Числове визначення відстані розташовується на середині лінії вимірювання, поруч з нею чи в спеціально відведеному місці.

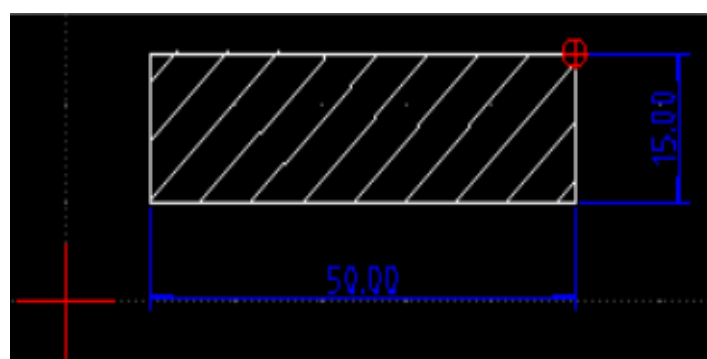


Рисунок 2.3 – Об'єкт з лінійними розмірами синього кольору

Систему ординаті виконують одну горизонтальну та одну вертикальну лінії виносу для встановлення спочатку для всього виду. Початок ідентифікується як 0,0, розміщений на кінцях цих рядків виносу.

Відстані вздовж осей  $x$  та  $y$  до інших точок на малюнку показуються за допомогою додаткових ліній виносу із відповідними числовими відомостями, розташованими належним чином.

Рисунках виготовляються в масштабі – недостатньо коли малюнок має той самий розмір, який і об'єкт, який він зображує (масштаб 1:1). Звичайний малюнок є меншим, а інженерні малюнки мають масштаби, такі як 1:2, 1:5 і так далі. У архітектурі традиційно використовується масштаб 1:20 для невеликих будинків. Дорожня карта може мати масштаб 1:100 000, що означає, що 1 см на карті дорівнює 1 км у природі. Інший екстремальний випадок - це об'єкти нано розмірів, які можуть мати масштаби, наприклад, 10 000:1 (також виражені як 1:0,0001).

DXF і DWG є основними форматами файлів у CAD. Обидва є форматами векторної графіки. Векторні зображення зберігають однакову якість зображення практично при будь-якому масштабі, що робить їх ідеальними для дизайну. Також легко редагувати окремі елементи, які формують векторне зображення, а також додавати і видаляти елементи.

DXF (скорочення від Data eXchange Format) був розроблений як інструмент для обміну проектами між різними CAD-програмами та векторно-орієнтованими програмами. Він зберігає 2D векторні зображення і може використовуватися практично всіма CAD-програмами, а також CNC (комп'ютерно-числове управління) та GIS (географічна інформаційна система) програмами. Кожен елемент малюнка викладено простим текстом або у форматі ASCII, який містить повний спектр буквено-цифрових символів. Завдяки їх характеристикам файлів, складні малюнки у форматі DXF стають великими – від кількох до сотень мегабайтів. При передачі такого документа його доводиться розбивати або стискати.

DWG (Drawing) є власним форматом від власника AutoCAD компанії Autodesk, Inc. Він може обробляти як інформацію у 2D, так і у 3D. Дані у форматі бінарних, тобто це серія 0 та 1. Зазвичай документ DWG є на 25% меншим, ніж документ DXF.

Як це зазвичай відбувається в галузі IT, власний формат DWG був реверсивно відібраний і випущений як OpenDWG Open Design Alliance не без юридичних суперечок із Autodesk, Inc. OpenDWG.com випущений під GPLv2, яку Лінус Торвальдс "Linux" вважає єдиною гідною ліцензією вільного та відкритого програмного забезпечення (FOSS).

Формат DWG все ще є експериментальним у LibreCAD.

АксонOMETрична проекція – це загальна назва методів проекції, які спрямовані на створення ілюзії тривимірного об'єкта на двовимірній поверхні.

АксонOMETрична проекція також відома як технічна проекція, оскільки вона використовується в технічному малюнку, включаючи CAD, і більш недавно в відео та комп'ютерних іграх. Вона відрізняється від лінійної проекції, використаної митцями, оскільки не має точок зникнення. АксонOMETрична проекція ближча до паралельної проекції, використаної в традиційних китайських живописах.

Існує три типи аксонOMETричної проекції: ізометрична, діаметрична і триметрична (рис. 2.4). Різниця можуть відчуватися тонкими, але є базові відмінності, особливо в тому, як три осі простору здаються скороченими.

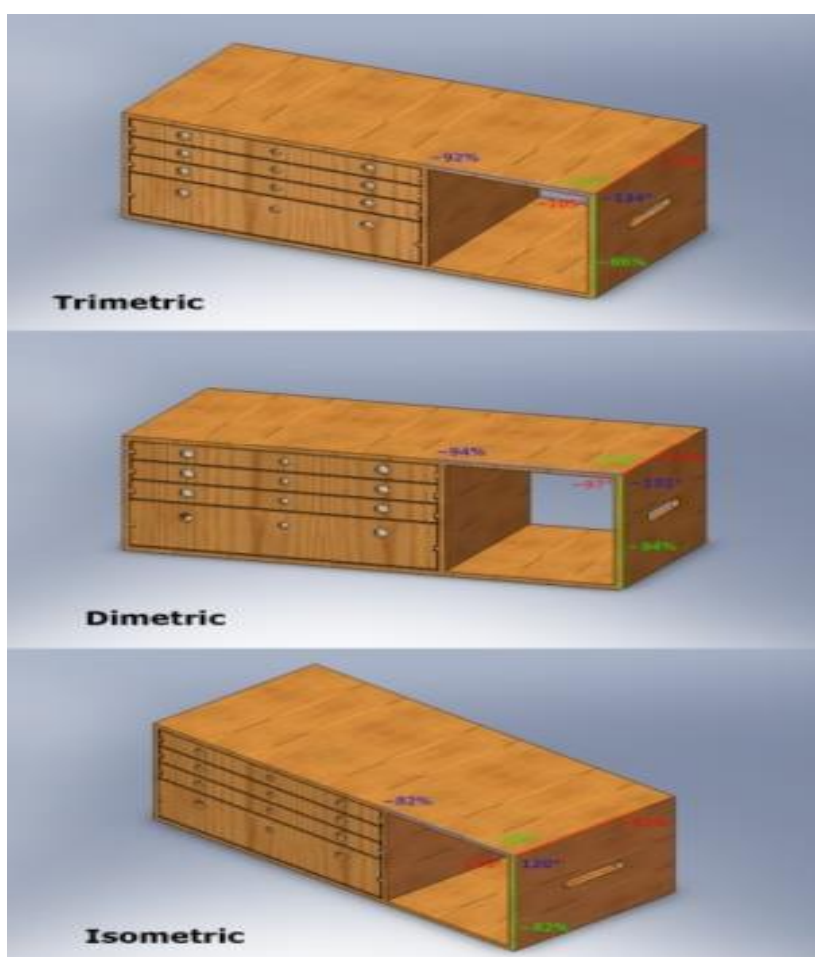


Рисунок 2.4 – Три типи аксонOMETричної проекції

У ізометричній проекції одиниця по осі однакова для всіх трьох осей (iso = рівний, рівномірний). Іншими словами, сантиметр – це сантиметр по осях x, y і z (з урахуванням тригонометричних рівнянь); отже, проекція називається "ізометричною". Діаметрична проекція, "di" = 2, використовує різні метрики для осей.

Ізометрична проекція пропонує переваги в інженерії та архітектурі завдяки легкості малювання через фіксовані кути  $120^\circ$  (або  $30^\circ$ , за бажанням), а також при необхідності брати виміри безпосередньо з рисунка. Однак ізометричні малюнки є лише візуальними посібниками, їх не слід використовувати як точні малюнки. Крім того, вони призводять до сприйняття спотворень, оскільки вони суперечать тому, як працює людський зір.

Варіант ізометричної проекції – це "військова проекція", яка використовує кути  $90^\circ$  ( $45^\circ$ ).

Кут ізометричної проекції викликає проблеми в комп'ютерних іграх, і з цієї причини використовується кут приблизно  $27^\circ$ .

Людський зір найкраще пристосований для обробки 3D-об'єктів у лінійному 2D проекція. Однак LibreCAD не призначений для лінійної проекції, але це має базові можливості, як це буде продемонстровано за допомогою невеликого тесту.

Line (Лінія): Команда для створення простих ліній. Введіть L або оберіть інструмент "Лінія". Команда яка дозволяє накреслити лінію при цьому указати дві точки для неї.

Circle (Коло): Команда для створення кола. Введіть C або оберіть інструмент "Коло". Команда завдяки якій можна накреслити коло указав центр кола та радіус цього кола.

Arc (Дуга): Команда для створення дуг. Введіть A або оберіть інструмент "Дуга". Команда яка дозволяє накреслити дугу указав дві точки та градус або кут розмаху яку потрібна для дуги.

Rectangle (Прямокутник): Команда для створення прямокутників. Введіть REC або оберіть інструмент "Прямокутник". Команда яка дозволяє накреслити прямокутник за допомогою двох точок.

Polyline (Мультисплайн): Команда для створення ламаних ліній. Введіть PL або оберіть інструмент "Мультисплайн". Команда яка дозволяє накреслити ламані лінії за допомогою N точок та градус або кут розмаху.

Text (Текст): Команда для введення тексту. Введіть T або оберіть інструмент "Текст". Команда яка дозволяє написати текст.

Dimension (Розмір): Команда для введення розмірів. Введіть D або оберіть інструмент "Розмір". Команда яка дозволяє вести розміри для інших команд.

Hatch (Штрихування): Команда для додавання штрихування або заливки областей. Введіть HATCH або оберіть інструмент "Штрихування". Команда яка дозволяє яка заштрихувати виділений об'єкт.

Trim (Обрізка): Команда для обрізки елементів. Введіть TR або оберіть інструмент "Обрізка". Команда яка дозволяє обрізати об'єкт.

Mirror (Відображення): Команда для відображення елементів відносно лінії або точки. Введіть MI або оберіть інструмент "Відображення". Команда яка дозволяє відобразити виділений об'єкт на в іншу сторону.

Move (Переміщення): Команда для переміщення елементів. Введіть MV або оберіть інструмент "Переміщення". Команда яка дозволяє перетаскувати об'єкт в інше місце.

Rotate (Обертання): Команда для обертання елементів. Введіть RO або оберіть інструмент "Обертання". Команда яка дозволяє обертати об'єкт по часовій стрільці за допомогою цього можна змінити її напрямком.

Offset (Зсув): Команда для створення паралельних ліній або кривих. Введіть O або оберіть інструмент "Зсув". Команда яка дозволяє створити за допомогою двох точок паралельні лінії або кривих.

LibreCAD має інтерактивний графічний інтерфейс користувача (GUI), також відомий як панель керування. Трошки часу займе щоб до нього звикнути, оскільки багато інструментів є незнайомими для тих, хто раніше не користувався CAD-програмами. Погляньте на зображення на наступній сторінці, воно показує GUI LibreCAD (v. 2.2.0rc1) у всій своїй красі: робоча область (також відома як робоча область) та основні набори інструментів, з якими ми працюємо. Є ще більше, деякі вікна приховані за кнопками, і вони, з свого боку, можуть мати безліч варіантів для вибору.

Графічний інтерфейс користувача (GUI) є досить особливим, лише рядок меню нагадує про більш звичайне програмне забезпечення. Панель інструментів має лише кілька знайомих піктограм, і багато з них мають зелений колір (рис. 2.5).

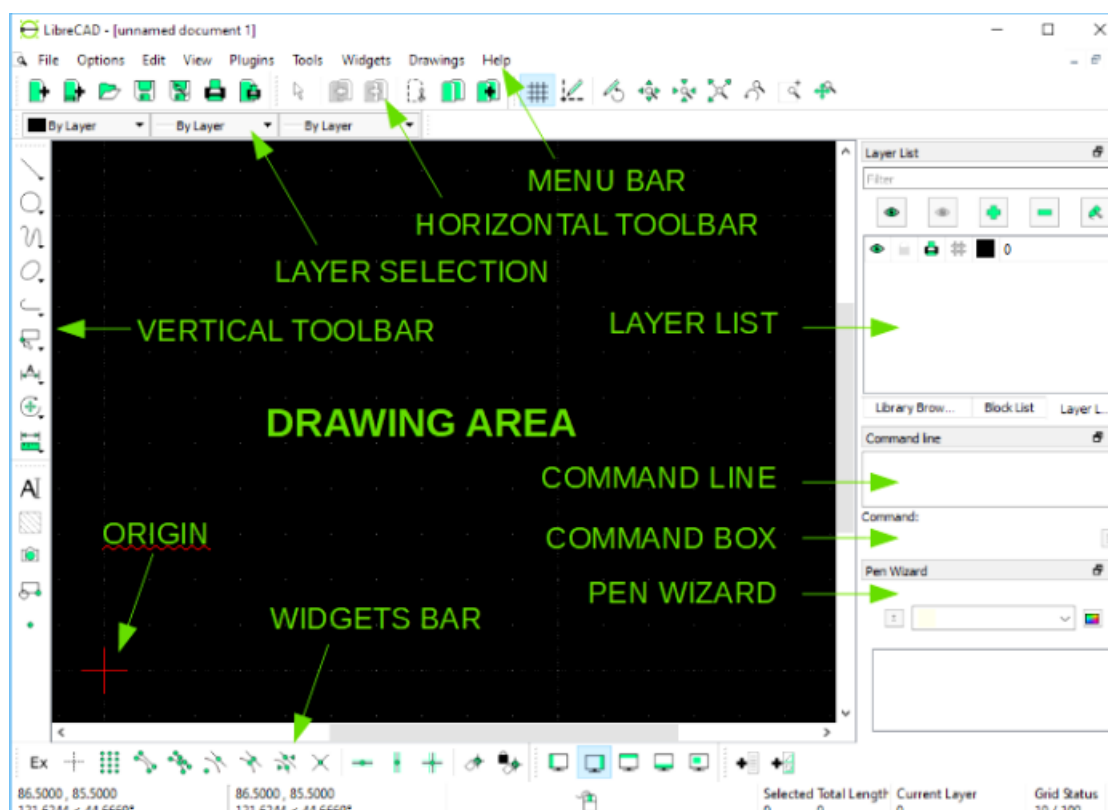


Рисунок 2.5 Інтерфейс LibreCAD

Все на панелі віджетів можна знайти на рядку меню, клацнувши Widgets > Панелі інструментів. Перша група в стандартному налаштуванні панелі віджетів – це інструменти Snap. Коли ви з'єднуєте лінії, завжди буде проміжок або перетин, неможливо попасти в точку чи лінію за допомогою миші – в кінцевому рахунку точка має нульові розміри. Рішенням є використання інструментів Snap.

Для більш докладного обговорення Snap, прочитайте документ "Snapping", який доступний за адресом.

Інструменти Snap дозволяють нам блокувати (приводити в крайню точку) віджети один до одного або до певних положень на екрані (рис. 2.6). Значки на панелі Snap є, зліва направо, Exclusive Snap Mode (Ex), Point Snap, Snap on Grid, Snap on Endpoint, Snap on Entity, Snap Center, Snap Middle, Snap Distance та Snap Intersection. Альтернатива, яку ми будемо використовувати найчастіше під час малювання це Snap on Grid (підкреслено).



Рисунок 2.6 – Значки на панелі Snap

Отримання текстових рядків у правильному положенні на малюнку може бути викликаним завданням. Моя думка говорить, що найкращим підходом є скасування всіх інструментів Snap, а потім перетягування тексту на його місце за допомогою миші – чи йдеться про MText чи Text.

Праворуч від панелі Snap розташовані ще чотири групи значків, перші дві з яких вважаються частиною панелі Snap. Спочатку йдуть три значки обмеження: обмежити горизонтально, обмежити вертикально та обмежити ортогональна. Далі два значки, які дозволяють встановлювати та блокувати віджети відносно нульового положення.

У третій групі з п'ятьма вікнами тільки два лівих мають помітний ефект, але це драматично. Ви можете побачити, що другий значок активний. Вимкніть його, і вікна списку шарів та командного рядка будуть закриті. Клацніть ліворуч, і палітра віджетів, показана вище праворуч, відкриється (закріплена) зліва від робочої області:

Функції закріплення можуть бути активовані або деактивовані, клацнувши Widgets > Панелі інструментів > Закріплення областей. Палітра містить значки для вертикальної панелі інструментів від Лінії до Інформації у трошки іншому порядку.

Проте є одна деталь (принаймні), якої бракує як у вертикальній панелі інструментів, так і в Палітрі віджетів порядок. Клацніть на Рядку меню Tools > Змінити > Порядок, і там у вас є альтернативи, за якими можна вибрати порядок відображення об'єктів (рис. 2.7).

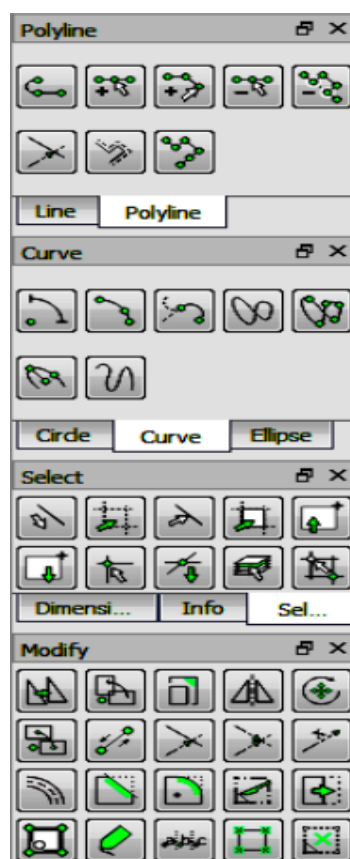


Рисунок 2.7 – Палітра віджетів

Остання група – це створювачі, кнопки створення меню та панелі інструментів.

Внизу панелі інструментів є відображення для отримання інформації в реальному часі про розташування вказівника миші (ліворуч) та віджетів, над якими ми працюємо (праворуч). Посередині є щось, що має надавати інформацію про кнопки миші, але це не працює.

Палітра віджетів – це лише одна палітра, яку можна закріпити зліва від Робочої області. На Рядку меню, якщо ми клацнемо Widgets > Закріпити віджети, є цілий набір альтернатив для закріплення, як показано вище ліворуч. Наприклад, я відкрив Палітру модифікацій (вона відкривається розгорнуто вздовж вертикальної панелі інструментів, тут я її стиснув).

Зверніть увагу, що є більше інструментів, які можна відкрити, клацнувши Widgets > Панелі інструментів. Які інструменти включати, вирішуєте ви в залежності від поточної ситуації.

В цьому розділі ми розглянули яка виглядає LibreCAD та його основний функціонал, інтерфейс та як ним користуватися.

### 2.3 Створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD

Для початку щоб створити програмні модулі для автоматизованого проектування деталі у системі LibreCAD нам потрібно побудувати блок схему та створити алгоритм роботи програми (рис. 2.8).

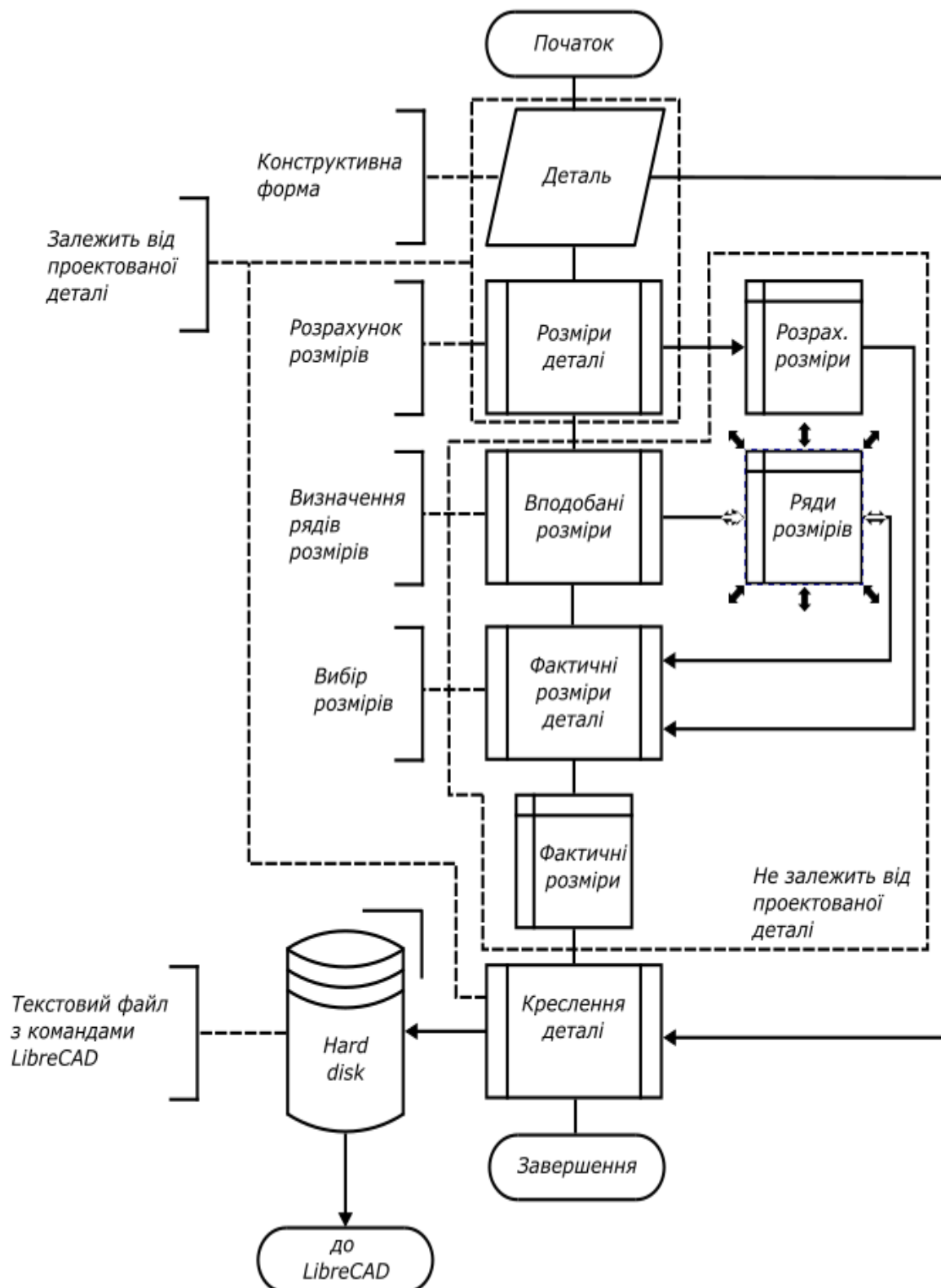


Рисунок 2.8 – Допрацьована блок схема

На рисунку зображена блок схема яка допоможе нам зрозуміти алгоритм програми та створити її. Де початок це – стартовий блок, з якого починається процес. Блоки, що відповідають за різні аспекти проектування деталі, включаючи:

- конструктивна форма;
- деталь;
- розрахункові розміри;

- визначення рядів розмірів;
- вибір розмірів.

Розміри деталі – пов'язані з фактичними вимірами частини. Фактичні розміри деталі – реальні розміри, взяті з моделі або задані в проекті. Блоки, пов'язані з введенням і коригуванням розмірів: вподобані розміри; ряд розмірів. Креслення деталі – фінальний етап, де візуалізуються розміри для створення креслення. Не залежить від проектованої деталі – це, вказує на елементи процесу, які є стандартними або не змінюються в залежності від конкретного проекту. Блок завершення, вказує на кінець процесу. Зовнішній блок Текстовий файл з командами LibreCAD підключений до "Hard disk" – це означає що, буде використання файлової системи для збереження інформації про проект.

Спочатку ми запускаємо програму після чого буде зроблені наступні дії. Після запуску програми ми починаємо робити деталь за допомогою різних формул. за допомогою чого користувач програми зможе отримати кращі числа для його варіанту деталі. Кращі числа будуть вибиратися способом округлення до найближчого кращого числа, які були вибрані за допомогою формули кращих чисел. Після чого програма починає отримувати розмір деталі та починається розрахунок розмірів, ці розрахунки будуть проводитися за допомогою формул які ми заздалегідь припишемо в кодї програми. Отримані розміри будуть переведені в кращі числа та перейдуть в ряди чисел. Після чого отримані ряди дадуть змогу нам отримати фактичні розміри це буде незалежно від проектованої деталі. Після буду розпочато креслення деталі за допомогою програми буде створений текстовий файл, який дасть можливість відкрити його за допомогою функціоналу командної строки LibreCAD. Так ми почнемо будувати деталь по одні із частин після чого буде проведена таж сама операція поти не накреслим цілу деталь.

Процес після завершення програми буде збережеться на жорсткий диск де буде знаходитися файл з текстовим документом, який був створений за допомогою різних формул та програми. В файлі буде знаходитися команди для побудови кожного елемента деталі. Далі цей файл буде відкрито за допомогою функціоналу LibreCAD куди буде перекинутий в командну строку. Після чого буде виконані поетапні кроки для побудови деталі.

Після написання блок схеми та алгоритмам програми буде написання тестового коду де буде проведено початкові тести програми.

Для початку розглянемо перший клас Sizes:

```
from math import pow, trunc
```

`trunc` є імпортом функцій `pow` та `trunc` з модулю `math`. Ця функція взята з модулю `math` і використовується для підняття числа до певного ступеня. Синтаксис: `pow(x, y)`, де `x` – це число, а `y` – ступінь.

`Trunc`. Ця функція також береться з модуля `math` і використовується для обрізання десяткової частини числа (залишається тільки ціла частина).

Отже, коли ви використовуєте `from math import pow, trunc`, ви можете використовувати ці функції без необхідності писати `math.` перед ними.

```
def PrefSizes_R5():
    val=[1.0,1.6,2.5,4.0,6.3,10.0]
    return val
```

Ця функція `PrefSizes_R5` повертає список перевизначених розмірів для ряду R5. У цьому випадку, розміри представлені у вигляді списку чисел.

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R5) можна викликати цю функцію за допомогою `PrefSizes_R5()` і отримати список з перевизначеними значеннями.

```
def PrefSizes_R10():
    val=[1.0,1.25,1.6,2.0,2.5,3.15,4.0,5.0,6.3,8.0,10.0]
    return val
```

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R10) можна викликати цю функцію за допомогою `PrefSizes_R10()` і отримати список з перевизначеними значеннями.

```
def PrefSizes_R20():
    val=[1.00,1.12,1.25,1.40,1.60,1.80,2.00,2.24,2.50,2.80,
        3.15,3.55,4.00,4.50,5.00,5.60,6.30,7.10,8.00,9.00,
        10.00]
    return val
```

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R20) можна викликати цю функцію за допомогою PrefSizes\_R20() і отримати список з перевизначеними значеннями.

```
def PrefSizes_R40():
    val=[1.00,1.06,1.12,1.18,1.25,1.32,1.40,1.50,1.60,1.70,
        1.80,1.90,2.00,2.12,2.24,2.36,2.50,2.65,2.80,3.00,
        3.15,3.35,3.55,3.75,4.00,4.25,4.50,4.75,5.00,5.30,
        5.60,6.00,6.30,6.70,7.10,7.50,8.00,8.50,9.00,9.5,
        10.00]
    return val
```

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R40) можна викликати цю функцію за допомогою PrefSizes\_R40() і отримати список з перевизначеними значеннями.

```
def PrefSizes_R80():
    val=[1.00,1.03,1.06,1.09,1.12,1.15,1.18,1.22,1.25,1.28,
        1.32,1.36,1.40,1.45,1.50,1.55,1.60,1.65,1.70,1.75,
        1.80,1.85,1.90,1.95,2.00,2.06,2.12,2.18,2.24,2.30,
        2.36,2.43,2.50,2.58,2.65,2.72,2.80,2.90,3.00,3.07,
        3.15,3.25,3.35,3.45,3.55,3.65,3.75,3.87,4.00,4.12,
        4.25,4.37,4.50,4.62,4.75,4.87,5.00,5.15,5.30,5.45,
        5.60,5.80,6.00,6.15,6.30,6.50,6.70,6.90,7.10,7.30,
        7.50,7.75,8.00,8.25,8.50,8.75,9.00,9.25,9.50,9.75,
        10.00]
    return val
```

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R80) можна викликати цю функцію за допомогою PrefSizes\_R80() і отримати список з перевизначеними значеннями.

```

def PrefSizes_R160():
    val=[1.000,1.015,1.030,1.045,1.060,1.075,1.090,1.105,1.120,1.135,
        1.150,1.165,1.180,1.190,1.220,1.230,1.250,1.265,1.280,1.300,
1.320,1.340,1.360,1.380,1.400,1.425,1.450,1.475,1.500,1.525,
        1.550,1.575,1.600,1.625,1.650,1.675,1.700,1.725,1.750,1.775,
        1.800,1.825,1.850,1.875,1.900,1.925,1.950,1.975,2.000,2.030,
        2.060,2.090,2.120,2.150,2.180,2.210,2.240,2.270,2.300,2.330,
        2.360,2.395,2.430,2.465,2.500,2.540,2.580,2.615,2.650,2.685,
        2.720,2.760,2.800,2.850,2.900,2.950,3.000,3.035,3.070,3.110,
        3.150,3.200,3.250,3.300,3.350,3.400,3.450,3.500,3.550,3.600,
        3.650,3.700,3.750,3.810,3.870,3.935,4.000,4.060,4.120,4.185,
        4.250,4.315,4.370,4.440,4.500,4.560,4.620,4.685,4.750,4.815,
        4.870,4.930,5.000,5.075,5.150,5.275,5.300,5.375,5.450,5.525,
        5.600,5.700,5.800,5.900,6.000,6.075,6.150,6.225,6.300,6.400,
        6.500,6.600,6.700,6.800,6.900,7.000,7.100,7.200,7.300,7.400,
        7.500,7.515,7.750,7.785,8.000,8.125,8.250,8.375,8.500,8.625,
        8,750,8.875,9.000,9.125,9.250,9.375,9.500,9.625,9.750,9.875,
        10.00]
    return val

```

Основна ідея полягає в тому, що ця функція може використовуватися для отримання перевизначених розмірів (в даному випадку, розмірів R160) можна викликати цю функцію за допомогою PrefSizes\_R160() і отримати список з перевизначеними значеннями.

```

def ChooseSize(size,PrefSizes,bigger=True):
    pf=PrefSizes
    n=len(pf)
    d=2
    while size<pf[0]:
        for i in range(n):
            pf[i]=pf[i]/10
        d=d+1

```

```

while size>pf[n-1]:
    for i in range(n):
        pf[i]=pf[i]*10
i=0
while not(pf[i]<=size and size<=pf[i+1]):
    i=i+1;
if bigger:
    res=round(pf[i+1],d)
else:
    res=round(pf[i],d)
return res

```

Ця функція ChooseSize призначена для вибору перевизначеного розміру на основі заданого розміру та списку перевизначених розмірів PrefSizes. Основний алгоритм полягає в знаходженні ближчого перевизначеного розміру, який може бути більшим чи меншим від заданого розміру, залежно від параметра bigger.

Ize параметр, який представляє собою заданий розмір. PrefSizes параметр, який представляє собою список перевизначених розмірів. Bigger параметр, який вказує, чи потрібно знаходити більший перевизначений розмір (True) чи менший (False). За замовчуванням шукається більший розмір.

Якщо заданий розмір менший за найменший перевизначений розмір, зменшуються всі розміри на порядок 10 і збільшується кількість десяткових знаків (d).

Якщо заданий розмір більший за найбільший перевизначений розмір, збільшуються всі розміри на порядок 10.

Потім використовується цикл для визначення інтервалу, в якому знаходиться заданий розмір, і вибору більшого чи меншого перевизначеного розміру в цьому інтервалі.

Функція повертає знайдений перевизначений розмір.

```

pf=PrefSizes_R40()
print(pf)
print(ChooseSize(0.0548,pf))
print(ChooseSize(4.6,pf))
print(ChooseSize(54.8,pf))

```

```
print(ChooseSize(346.8,pf))
```

Цей код буде використаний для визначення перевизначеного розміру для заданого значення у інженерних розрахунках.

Далі ми розглянемо наступний клас `standard`

```
st1_sizesnames=("s1","s2","s3","s4","s5","s6")
st1_sizes={1:(1,2,3,4,5),
           2:(2,4,6,8,10),
           3:(3,6,9,12,15)
           }
```

`st1_sizesnames`: Це кортеж, що містить назви розмірів, такі як "s1", "s2", "s3", "s4", "s5" і "s6".

`st1_sizes`: Це словник, де ключі – це числа (1, 2, 3), а значення – це кортежі, які містять розміри для кожного ключа.

```
def standard_sizes(keysize,st_sizes,st_sizesnames):
    if keysize in st_sizes:
        sizes={st_sizesnames[0]:keysize}
        for i in range(len(st_sizes[keysize])):
            sizes.setdefault(st_sizesnames[i+1],st_sizes[keysize][i])
        return sizes
    else:
        return False
```

Ця функція `standard_sizes` призначена для створення розмірів словника на основі ключа `keysize`, розмірів словника `st_sizes` та назв розмірів `st_sizenames`. Функція перевіряє, чи існує заданий ключ у словнику `st_sizes`, і якщо так, то створює словник, використовуючи назви розмірів і відповідні значення з `st_sizes[keysize]`. `Keysize` це ключ, який ви шукаєте у словнику `st_sizes`.

`St_sizes` словник, що містить розміри для кожного ключа.

St\_sizesnames кортеж, що містить назви розмірів, які будуть використовуватися як ключі у новому словнику.

Перевірка, чи існує keysize у st\_sizes.

Якщо так, створення нового словника sizes, де st\_sizesnames є ключем, а keysize - значенням цього ключа.

Далі цикл використовується для додавання інших елементів у словник sizes на основі st\_sizesnames та відповідних розмірів з st\_sizes[keysize].

Якщо ключ не існує у st\_sizes, функція повертає False

```
print(standard_sizes(1,st1_sizes,st1_sizesnames))
print(standard_sizes(2,st1_sizes,st1_sizesnames))
```

Код викликає функцію standard\_sizes для двох різних ключів (1 і 2) з використанням наведених словників та кортежа. Результати виводяться на екран за допомогою функції print.

Далі розглянемо клас draw.

```
sizes={"size1":16,"size2":16,
      "size3":24
      }
```

У цьому словнику є три розміри: "size1" зі значенням 16, "size2" також зі значенням 16, і "size3" зі значенням 24. Ви можете використовувати цей словник для представлення розмірів у вашому коді або проекті.

```
def draw_part(sizes,filename):
    s1=sizes["size1"]
    s2=sizes["size2"]
    s3=sizes["size3"]

    file=open(filename+".txt","w")
    x1=0; y1=0; p1=str(x1)+","+str(y1)
    file.write("p1="+p1+"\n")
    x2=s1; y2=s2; p2=str(x2)+","+str(y2)
```

```

file.write("p2="+p2+"\n")
file.write("li;\p1;\p2\n"); file.write("k\n")
x3=x2+10; y3=x2+5; p3=str(x3)+","+str(y3)
file.write("p3="+p3+"\n")
file.write("li;@0,0;\p3\n"); file.write("k\n")
file.write("arc;\p1;\p2;\p3\n"); file.write("k\n")
file.write("ci;\p3;15\n"); file.write("k\n")
file.close

```

```
draw_part(sizes,"testdraw")
```

Функція `draw_part` призначена для створення текстового файлу, який містить команди для зображення частини зазначених розмірів. Основні етапи цієї функції.

Отримання розмірів іменованих "size1", "size2" і "size3" з переданого словника `sizes`.

Відкриття файлу: створюється текстовий файл з ім'ям, яке формується за допомогою з'єднання переданого імені файлом (`filename`) з розширенням ".txt".

Запис координат: записуються координати точок `p1`, `p2`, `p3` у текстовий файл у вигляді рядків з координатами.

Запис команд для зображення: записуються команди для з'єднання точок лінією (`li`) та кругової дуги (`arc`). Також додається команда для зображення кола (`ci`).

Закриття файлу: файл закривається.

## 2.4 Висновок методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD

В ході дослідження методу створення програмних модулів для автоматизованого проектування деталей у системі LibreCAD було вибрано мову програмування Python який краще підходить для цієї задачі. Порівняли Python з іншими мовами та розглянули кожну із них.

Був проведений аналіз формул та теоретичні основи автоматизованого проектування деталей. Проведено аналіз термінів та було вибрано формули які можуть знадобитися. Після чого було розглянено засоби системи LibreCAD для

автоматизованого проектування деталей. Було розглянуто команди та інтерфейс LiberCAD та його можливості.

Також створили програмні модулі для автоматизованого проектування деталей у системі LibreCAD написали алгоритм програми та побудували блок схему, написали тестовий код програми де дослідили частину коду. Також підготувалися до написання та тестування програмних модулів.

### 3 ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ДЕТАЛЕЙ З ВИКОРИСТАННЯМ ПРОГРАМНИХ МОДУЛІВ

Для того щоб дослідити автоматизоване проектування деталей з використанням програмних модулів нам потрібно написати програму. В якій ми зможемо порівняти та дослідити різні аспекти, а саме залежність деталі від переважних розмірів, порівнюючи різницю між найменшим розкидом та самим великим. Також потрібно дослідити різницю у розмірі. Також розглянути чи зміниться об'єм. Та дослідити розміри.

#### 3.1 Вплив вибору ряду переважних розмірів на результат проектування

Проведемо дослідження перетворення деталі завдяки вибору переважних розмірів. Для цього нам потрібно написати код який виведе графік. Для цього скористаймося вже готовим кодом який ми зробили раніше да додаймо класи та за допомогою них зробимо дослідження.

Для початку імпортуємо бібліотеку `matplotlib.pyplot` яка дозволяє нам створювати графіки та діаграми в Python. Зазвичай ця бібліотека використовується для візуалізації даних.

```
import matplotlib.pyplot as plt
```

Після чого пишемо код який визначає функцію `create`, яка створює базовий графік за допомогою бібліотеки `matplotlib.pyplot`. Створюємо фігури та осі: використовуючи `plt.subplots()`, функція створює фігуру та об'єкт осі, які потім можна використовувати для створення графіків.

Встановлюємо заголовка та підписів осей: використовуючи методи `set_title`, `set_xlabel`, і `set_ylabel`, встановлюються заголовок графіка та підписи осей.

Встановлюємо сітки: використовуючи метод `ax.grid`, встановлюється сітка графіка з заданими параметрами.

Додаємо легенди: якщо параметр `legend` встановлений в `True`, то викликається метод `ax.legend()`, що додає легенду до графіка. Повертаємо об'єкту осі: функція повертає об'єкт осі (`ax`), який можна використовувати для додавання графіків та інших елементів до поточного графіка.

Ця функція слугує як вихідна точка для створення графіків у нашому коді та може бути розширена залежно від потреб.

```
def create(title, titlex, titley, legend = True):
    fig, ax = plt.subplots()
    ax.set_title(title)
    ax.set_xlabel(titlex)
    ax.set_ylabel(titley)
    ax.grid(visible=True, color='k', linestyle='-', linewidth=0.5)
    if legend:
        ax.legend()
    return ax
```

Далі пишемо функцію `append` яка використовується для додавання нового графіку до існуючого графіка. Вона використовує метод `plot` бібліотеки `matplotlib.pyplot`, щоб намалювати лінію або точки на графіку за допомогою заданих значень `x` і `y`. Ця функція очікує три аргументи:

- `chart`: об'єкт осі, до якого буде додано новий графік;
- `x`: значення по осі `X`;
- `y`: значення по осі `Y`;
- `label`: підпис для легенди (необов'язковий параметр).

```
def append(chart,x,y,label):
    chart.plot(x,y,label=label)
```

Після чого пишемо функція `legend` яка використовується для відображення легенди на графіку. Вона викликає метод `legend` для об'єкта осі `chart`. Це дозволяє показати легенду для кожного графіку, який був доданий до осі. Ця функція очікує один аргумент `chart`: об'єкт осі, для якого буде відображена легенда.

```
def legend(chart)
    chart.legend()
```

Далі пишемо таку функцію `CurvesPlot` яка використовується для створення графіка з кривими, які передаються у вигляді списку `Curves`.

Кожна крива представляє собою внутрішній список, що містить ім'я кривої та параметри для її відображення. Після цього ви вводимо параметри функції для `Curves`: Список, що містить заголовок графіка, підписи осей та список кривих. `title`, заголовок графіка. `xlabel`: підпис для осі X. `Ylabel`; підпис для осі Y. Створюємо графік за допомогою `fig, ax = plt.subplots()`.

Створення фігури та об'єкта осі для графіка. Пишемо заголовки та підписи осей `ax.set_title(Curves[0])`: Встановлення заголовка графіка. `ax.set_xlabel(Curves[1])`: встановлення підпису для осі X. `ax.set_ylabel(Curves[2])`: встановлення підпису для осі Y. Додаємо криві `for curve in Curves[3]:`: Ітерація по списку кривих. `ax.plot(...)`: Додавання кривої до графіка з параметрами, переданими через `curve[0]` (`label`, `color`, `linestyle`, `linewidth`, `marker`, `markersize`) та даними для осей X та Y через `curve[1]` та `curve[2]`

Та встановлюємо сітки та легенди `ax.grid(...)`: Встановлення сітки графіка. `ax.legend()`: виведення легенди для кривих.

```
def CurvesPlot(Curves):
    fig,ax=plt.subplots()
    ax.set_title(Curves[0])
    ax.set_xlabel(Curves[1])
    ax.set_ylabel(Curves[2])
    for curve in Curves[3]:
        ax.plot(curve[1],curve[2],
                label=curve[0][0],
                color=curve[0][1],
                linestyle=curve[0][2],
                linewidth=curve[0][3],
                marker=curve[0][4],
                markersize=curve[0][5])

    ax.grid(visible=True,color='k', linestyle='-', linewidth=0.5)
    ax.legend()
```

Зробимо далі клас `Research_1` який буде малювати графіки в якому ми спочатку будемо посилатися на інші класи.

```
import sizes
import numpy as np
import matplotlib.pyplot as plt
import ChartsPlot
```

Потім ми ініціалізуємо змінні `R5`, `R10`, `R40`, `R80`, `R160` із використанням функцій `PrefSizes_R5()`, `PrefSizes_R10()`, `PrefSizes_R40()`, `PrefSizes_R80()`, `PrefSizes_R160()` від модуля `sizes`.

```
R5 = sizes.PrefSizes_R5()
R10 = sizes.PrefSizes_R10()
R40 = sizes.PrefSizes_R40()
R80 = sizes.PrefSizes_R80()
R160 = sizes.PrefSizes_R160()
```

Використовуємо бібліотеку NumPy для створення масиву розмірів `SIZES` та інших масивів `SIZESR5`, `SIZESR10`, `SIZESR40`, `SIZESR80`, `SIZESR160` для відповідних значень розмірів.

Задача цього фрагмента коду полягає в тому, щоб для кожного значення в `SIZES` визначити переважний розмір (preferred size) відповідно до вказаних рядів (`R5`, `R10`, `R40`, `R80`, `R160`). Для цього використовується функція `ChooseSize()` з модуля `sizes`.

Після виконання цього коду буде п'ять масивів (`SIZESR5`, `SIZESR10`, `SIZESR40`, `SIZESR80`, `SIZESR160`), які представляють вибрані переважні розміри відповідно до рядів `R5`, `R10`, `R40`, `R80`, `R160` для відповідних значень в масиві `SIZES`. Ці дані можна використовувати для подальших обчислень або аналізу.

```
number = 1000
SIZES = np.linspace(1.0, 10.0, num=number)
SIZESR5 = np.zeros(number)
SIZESR10 = np.zeros(number)
SIZESR40 = np.zeros(number)
SIZESR80 = np.zeros(number)
SIZESR160 = np.zeros(number)
```

```

for i in range(number):
    SIZESR5[i] = sizes.ChooseSize(SIZES[i], R5)
    SIZESR10[i] = sizes.ChooseSize(SIZES[i], R10)
    SIZESR40[i] = sizes.ChooseSize(SIZES[i], R40)
    SIZESR80[i] = sizes.ChooseSize(SIZES[i], R80)
    SIZESR160[i] = sizes.ChooseSize(SIZES[i], R160)

```

Створюємо графік (chart1) з використанням функції create() з модуля ChartsPlot. Потім додаєте криві на цей графік за допомогою функції append(), яка додає дані для кожної кривої. У вас є криві для рядів R5, R10, R40, R80, R160.

Нарешті, викликаєте функцію legend() для відображення легенди на графіку, яка вказує, яка крива відповідає якому ряду.

Цей код буде корисним для візуалізації, порівняння та аналізу впливу вибору ряду переважних розмірів на визначені розміри в зазначеному діапазоні.

```

chart1 = ChartsPlot.create("від 1 до 10", "визначений розмір", "переважний розмір")
ChartsPlot.append(chart1, SIZES, SIZESR5, "R5")
ChartsPlot.append(chart1, SIZES, SIZESR10, "R10")
ChartsPlot.append(chart1, SIZES, SIZESR40, "R40")
ChartsPlot.append(chart1, SIZES, SIZESR80, "R80")
ChartsPlot.append(chart1, SIZES, SIZESR160, "R160")
ChartsPlot.legend(chart1)

```

Далі ми робимо теж саме але змінюємо діапазон переважний розмір.

Спочатку проведемо аналіз процесу та коду для створення автоматизованих модулів дизайну в системі LibreCAD. У ньому визначено алгоритм програми, функції та класи, які використовуються для різних завдань, таких як визначення бажаних розмірів для ряду розмірів (R5, R10, R20, R40, R80, R160), а також те, як вибирати розміри на основі цих уподобань. Також детально описаний процес малювання валів із визначеними розмірами та генерації текстового файлу з командами для LibreCAD для створення малюнків.

Тепер давайте проаналізуємо графіки. Графік "від 1 до 10" (рис. 3.1). Здається, що цей графік показує бажані розміри для визначеного діапазону розмірів від 1 до 10 згідно з різними серіями (R5, R10, R40, R80, R160). Кожна лінія представляє серію, з х-осю, що

вказує визначений розмір, і у-осю, яка показує бажаний розмір. Ступінчастість ліній вказує на те, що при збільшенні визначеного розміру бажаний розмір стрибає на наступний стандартний розмір у серії.

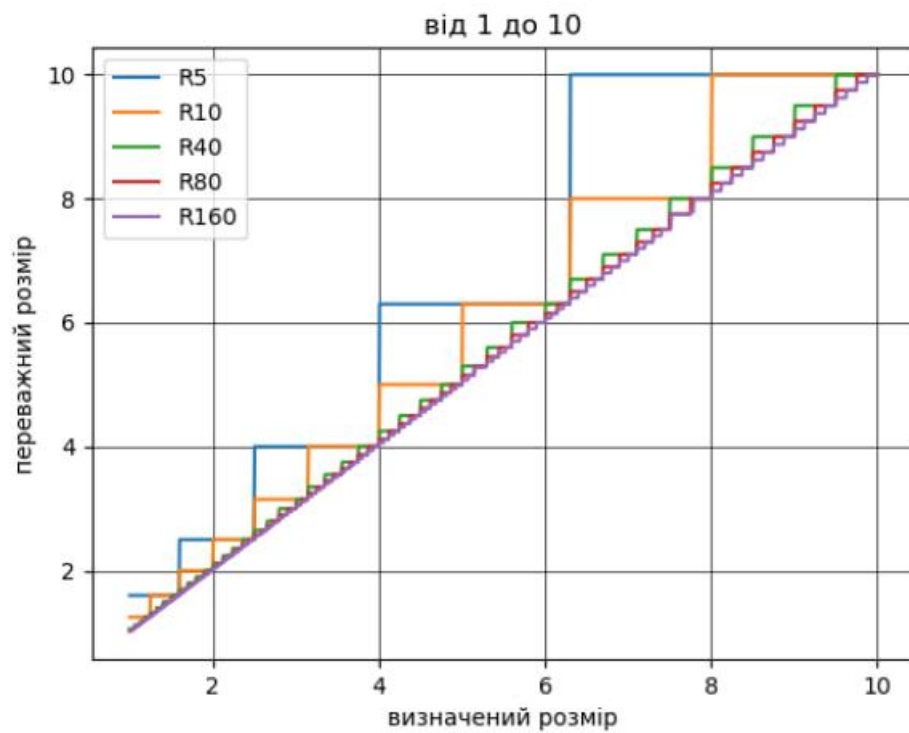


Рисунок 3.1 – Графік від 1 до 10

Графік "від 10 до 20" (рис. 3.2). Схожий на перший графік, але для діапазону розмірів від 10 до 20. Використовуються ті ж самі серії, і ми можемо спостерігати, що для більших розмірів стрибки в бажаних розмірах стають більшими, відображаючи логарифмічний характер стандартних серій.

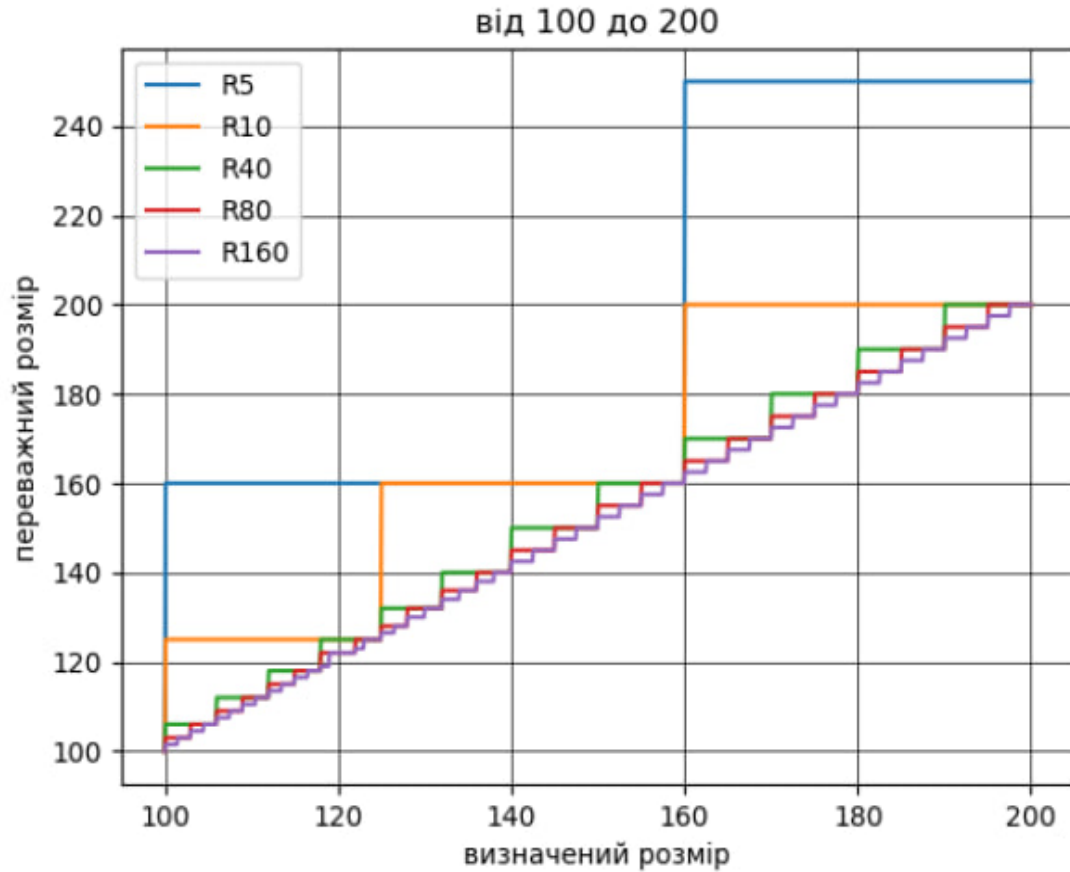


Рисунок 3.2 – Графік від 10 до 20

Графік "від 100 до 200" (рис. 3.3). Цей графік розширює діапазон від 100 до 200. Він слідує тому самому шаблону, що й попередні графіки, і показує, що бажані розміри збільшуються ступінчасто при збільшенні визначених розмірів.

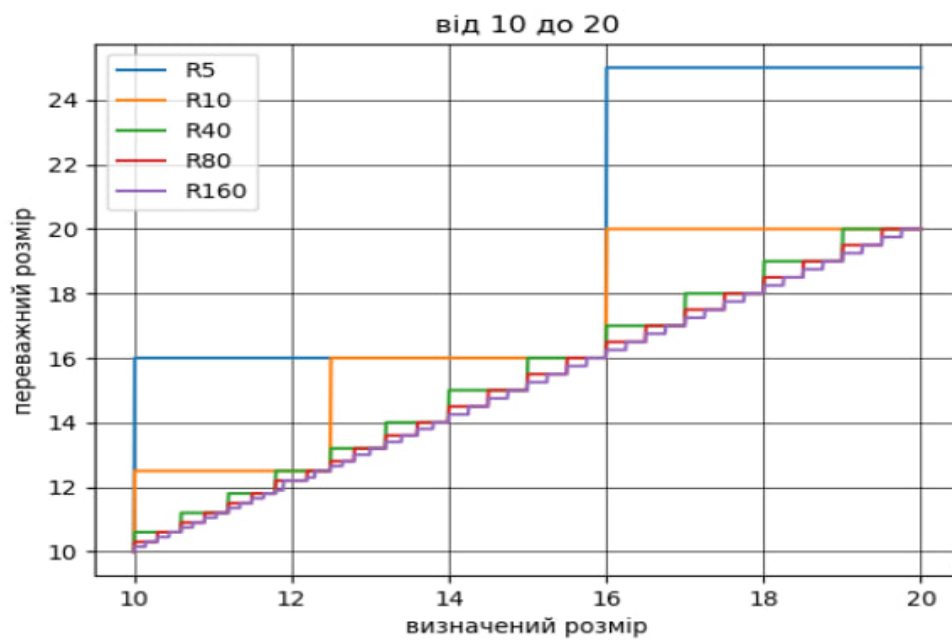


Рисунок 3.3 – Графік від 10 до 20

Графік "від 2000 до 3000" (рис. 3.4). Цей графік представляє бажані розміри для набагато ширшого діапазону, від 2000 до 3000. Незважаючи на більше значення, патерн є консистентним із попередніми графіками, що вказує на масштабованість вибору бажаних розмірів у широкому діапазоні розмірів.

Ці графіки корисні для розуміння того, як вибираються бажані розміри для різних компонентів в інженерному дизайні і можуть бути корисні при виборі стандартних розмірів для деталей для відповідності промисловим нормам і стандартам.

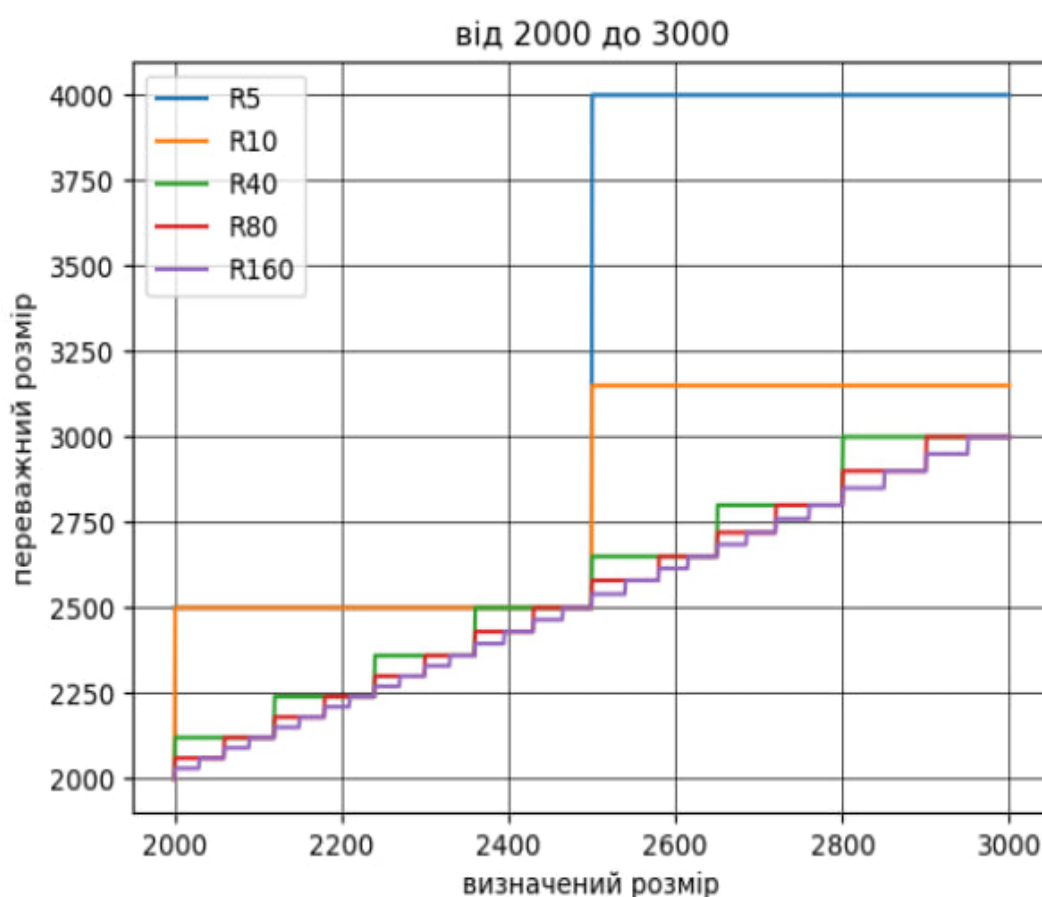


Рисунок 3.4 – Графік від 100 до 200

Основна функціональність полягає в автоматизованому виборі бажаних розмірів для компонентів на основі визначених уподобань та стандартів. Зокрема, модуль дозволяє вибирати розміри із певних серій (R5, R10, R20, R40, R80, R160) для різних діапазонів значень.

Графіки, які подані у кодї, ілюструють логіку вибору бажаних розмірів для різних діапазонів значень. За їхніми результатами можна зрозуміти, як програма визначає оптимальні розміри з урахуванням стандартів та уподобань інженера.

Отже, програмний модуль в LibreCAD, який використовує цей код, може ефективно полегшити процес вибору розмірів для конкретних компонентів у проектах інженерного дизайну, забезпечуючи відповідність стандартам і забезпечуючи швидкість та точність у визначенні розмірів.

### 3.2 Можливості використання типових стандартних рішень щодо проектування деталей

Проведемо дослідження можливості використання типових стандартних рішень щодо проектування деталей. Для цього скористаймося вже готовим кодом який ми зробили раніше та додаймо клас за допомогою якого зробимо дослідження та проаналізуємо результати.

Почнемо писати код для початку викликаєм 2 модулі за допомогою яких дамо класу змогу брати за основу клас sizes та CylEndShafts.

```
import sizes, CylEndShafts
```

Далі пишемо функцію призначена для створення текстового опису геометрії валу з вказаними параметрами D (діаметр), L (довжина), x (початкова координата x), y (початкова координата y) у файлі. Файл використовується в форматі який буде інтерпретований LibreCAD для створення графічних зображень. Координати початкової точки (p1), кінцевої точки (p2), та двох інших точок (p3, p2 після обертання) обчислюються залежно від введених параметрів. Координати кожної точки записуються у вигляді рядків у файл. Записуються команди для з'єднання точок лініями (li). Це план для побудови геометричних форм в LibreCAD. Додамо команди для завершення опису геометрії у файл.

```
def draw_shaft(D,L,x,y,file):
    x1=x; y1=y+D/2; p1=str(x1)+","+str(y1); file.write("p1="+p1+"\n")
    x2=x+L; y2=y1; p2=str(x2)+","+str(y2); file.write("p2="+p2+"\n")
    file.write("li;\p1;\p2\n"); file.write("k\n")

    x3=x2; y3=y2-D; p3=str(x3)+","+str(y3); file.write("p3="+p3+"\n")
    file.write("li;\p2;\p3\n"); file.write("k\n")
```

```
x2=x3-L; y2=y3; p2=str(x2)+","+str(y2); file.write("p2="+p2+"\n")
file.write("\li;\p3;\p2\n"); file.write("k\n")
file.write("\li;\p2;\p1\n"); file.write("k\n")
```

Далі код буде запитує вас про деякі параметри валу, включаючи вибір набору переважних розмірів (R5, R10 і т. д.), обчислений діаметр валу (D), розмір краю валу (SE), довжину валу без країв (L), ім'я файлу для збереження зображення валу. Якщо користувач не ввів нічого, обирається значення R160 як значення за замовчуванням. Програма використовує функцію ChooseSizeR з модуля sizes, яка ймовірно визначає переважний розмір відповідно до введеного розміру та набору розмірів R. Також викликаємо функцію standard\_edge з модуля CylEndShafts для отримання переваженого розміру краю валу. Далі користувач використовує функція standard\_edge з модуля CylEndShafts, щоб визначити та вивести переважні розміри краю валу. Далі проходить виведення переважного діаметр валу. Аналогічно діаметру, користувач вводить обчислену довжину валу, а потім викликається функція ChooseSizeR для визначення переважного розміру довжини. Далі проходить виведення переважної довжини валу. Після чого користувач вводить ім'я файлу для збереження зображення валу. Далі проходить перевірка на порожній ввід і встановлення значення за замовчуванням, якщо ввід порожній. Якщо користувач не ввів ім'я файлу, використовується значення "shaft" як значення за замовчуванням. Додаємо розширення ".txt" до імені файлу. Це гарантує, що файл буде збережено у форматі текстового файл.

```
print("premissible notions for the preferenced sizes set are: R5, R10, R20, R40, R80,
R160")
R=input("input notions of the preferenced sizes set (R5,R10,...): ")
if R == "":
    R="R40"
D=float(input("input computed diameter of the shaft, mm: "))
D=sizes.ChooseSizeR(D,R,bigger=True)
print("preferenced diameter of the shaft: "+str(D)+" mm")
SE=CylEndShafts.standard_edge(D);
print("preferenced sizes of the shaft edge (mm):")
print(SE)
```

```

L=float(input("input computed length of the shaft without edges, mm: "))
L=sizes.ChooseSizeR(L,R,bigger=True)
print("preferenced length of the shaft without edgest: "+str(L)+" mm")
filename=input("input file name to save the draw of the shaft: ")
if filename == "":
    filename="shaft"
filename=filename+".txt"

```

Далі фрагмент коду відкриває файл для запису (у режимі "w"), рисує вал, а потім викликає функції для малювання розмірів краю валу на обох кінцях валу, та закриває файл. Спочатку відбувається відкриття файлу з ім'ям filename у режимі запису ("w"). Це дозволить записати в файл. Далі встановлюються початкові координати ( $x=0$ ,  $y=0$ ), і потім викликається функція draw\_shaft, яка малює вал з заданими параметрами (D і L). Після чого відбуваються виклик функції st1208066I\_sizes1\_draw для малювання розмірів краю валу на лівому кінці валу. А далі код починаючи з кінцевого значення x (L), викликається та малюється функція st1208066I\_sizes1\_draw для розмірів краю валу на правому кінці. Після завершення запису файл закривається.

```

file=open(filename,"w")
x=0; y=0; draw_shaft(D,L,x,y,file)
CylEndShafts.st1208066I_sizes1_draw(x,y,SE,file,rightedge=False)
x=L;
CylEndShafts.st1208066I_sizes1_draw(x,y,SE,file,rightedge=True)
file.close

```

Та на завершення рядок коду який просто чекає, доки користувач натисне клавішу Enter, і тоді програма завершує свою роботу. Це зроблено для того, щоб утримати вікно програми від закриття відразу після виконання всіх дій. Після того, як користувач натисне Enter, програма буде завершена (рис. 3.5).

```

input("press enter to exit program")

```

```

premissible notions for the preferenced sizes set are: R5, R10, R20, R40, R80, R160
input notions of the preferenced sizes set (R5,R10,...): R40
input computed diameter of the shaft, mm: 54.5
preferenced diameter of the shaft: 56.0 mm
preferenced sizes of the shaft edge (mm):
{'d': 50.0, 'l': 110.0, 'r': 2.5, 'c': 2.0}
input computed length of the shaft without edges, mm: 317.5
preferenced length of the shaft without edges: 335.0 mm
input file name to save the draw of the shaft: R40Test

```

Рисунок 3.5 – Консоль програми

Після написання коду ми почнемо дослідження, а саме розглянемо всі переважні розміри такі як R5, R10, R20, R40, R80, R160. Виберемо один діаметр для всіх переважних розмірів це буде 54,5 мм, а також виберемо переважний розмір довжини це буде 317,5 мм.

Для початку почнемо з R5 в програмі введемо ряд R5 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR5 після чого завантажимо цей файл у LibreCAD (рис. 3.6). Так як переважний розмірів R5 та R10.

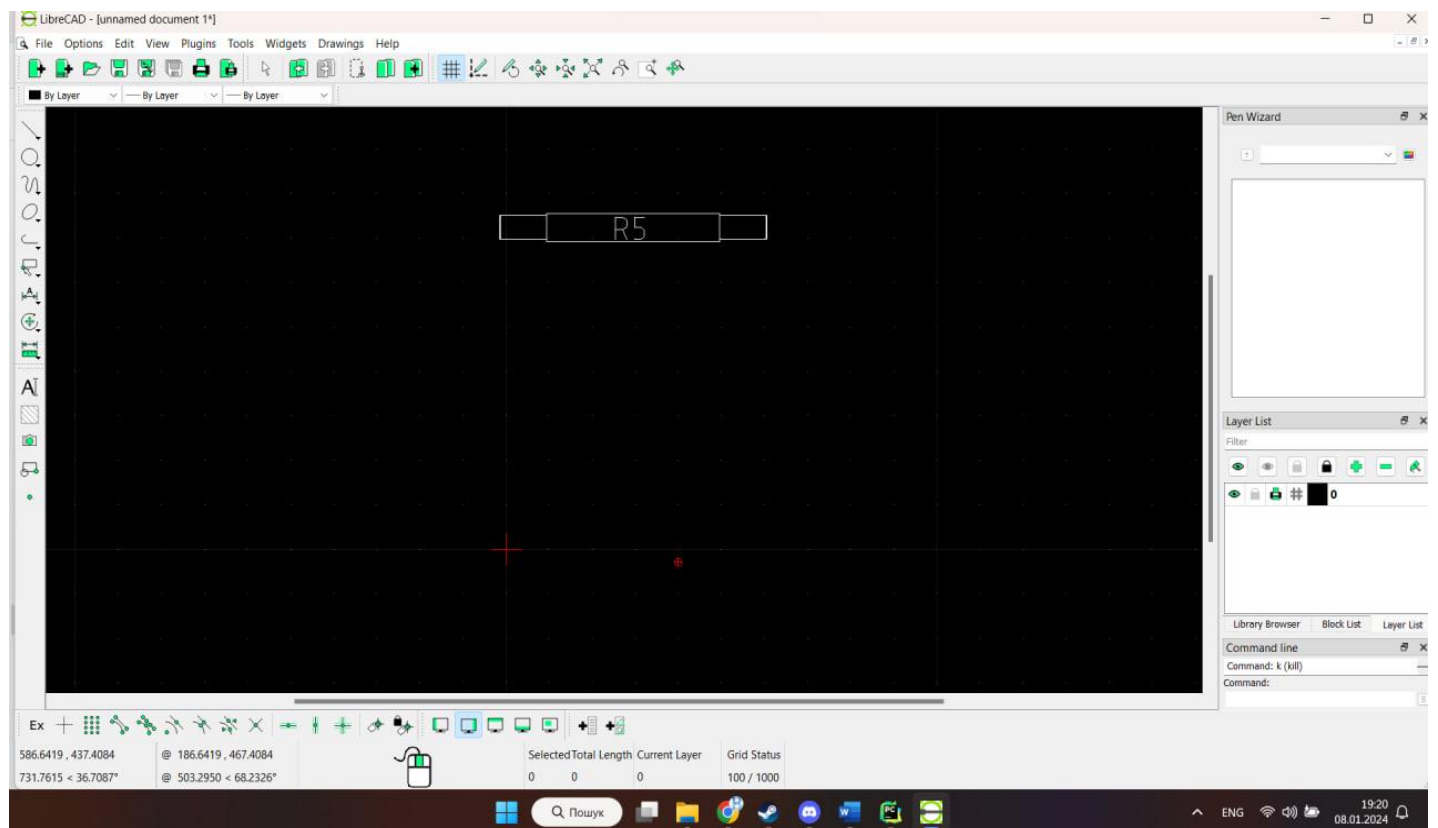


Рисунок 3.6 – Дослідження файлу ShaftR5

Після завантаження R5 в LibreCAD зробимо наступне в програмі введемо ряд R10 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR10 після чого завантажимо цей файл у LibreCAD.

Як видно на рисунку при введенні ряду R10 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм після чого завантажили цей файл у LibreCAD (рис. 3.7).

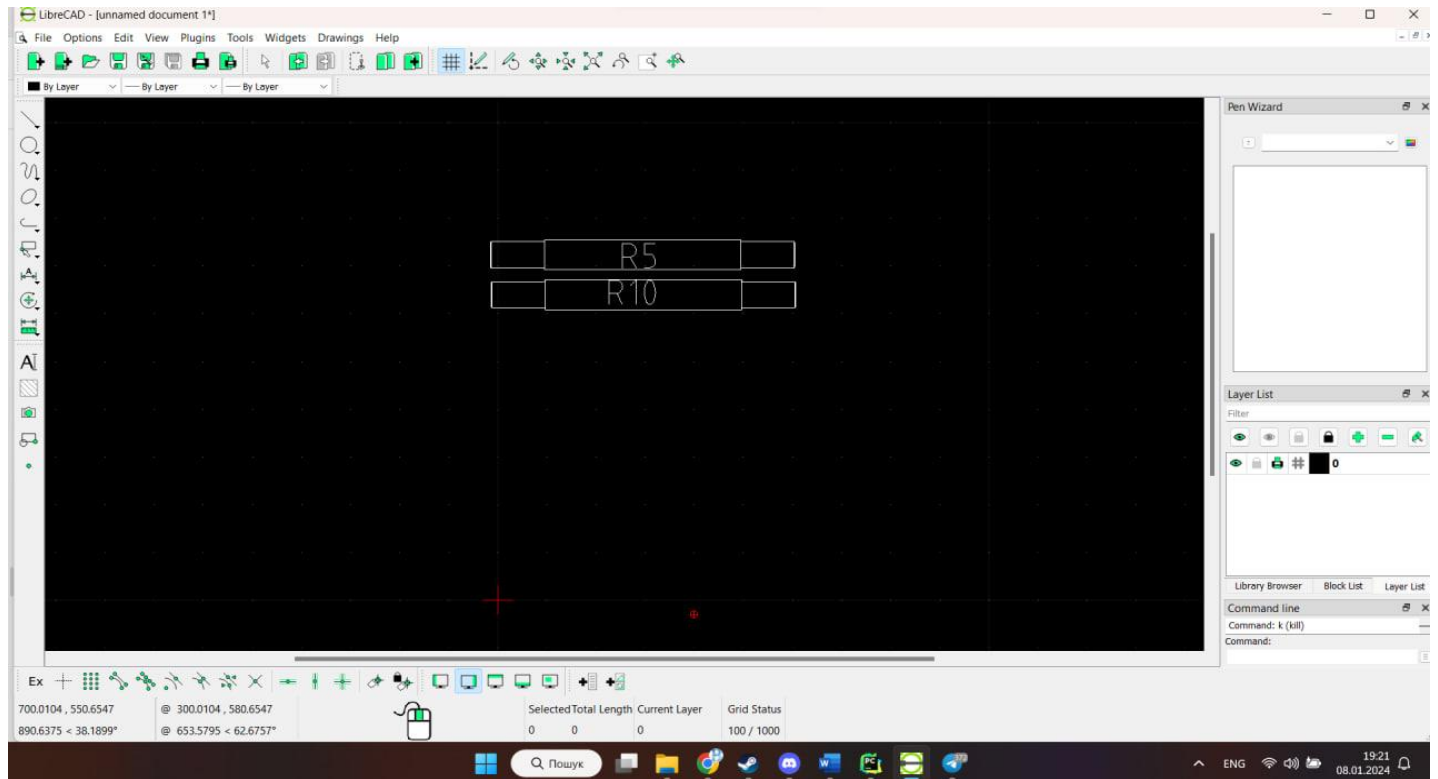


Рисунок 3.7 – Дослідження файлу ShaftR10

Так як R5 має [1.0, 1.6, 2.5, 4.0, 6.3, 10.0] переважний ряд розмірів, а R10 [ 1.0, 1.25, 1.6, 2.0, 2.5, 3.15, 4.0, 5.0, 6.3, 8.0, 10.0], то і R5 та R10 будуть видавати однакові розміри такі як діаметр 63 мм та довжину 400 мм.

Далі так як минулого разу ми проведемо вже дослідження с порівнянням переважні ряди розмірів R5, R10 та новий R20 Для цього нам потрібно знову запустити програму ввести ряд R20 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR20 після чого завантажимо цей файл у LibreCAD. Після чого ми отримали (рис. 3.8). Тепер наглядно видно що ми отримали іншу деталь по розміру це сталося тому що переважні ряди розмірів R5 та R10 збіглися для розміру діаметр 54,5 мм та розміру довжини 317,5 мм. Так як переважний ряд розміру R20 [1.00, 1.12, 1.25, 1.40, 1.60, 1.80, 2.00, 2.24, 2.50, 2.80, 3.15, 3.55, 4.00, 4.50, 5.00, 5.60, 6.30, 7.10, 8.00, 9.00, 10.00] при вводі розмір діаметр 54,5 мм, а розмір довжини 317,5 мм виходить що переважний розмір діаметру 56 мм, а переважна довжина 355 мм, тому деталь стала меншою.

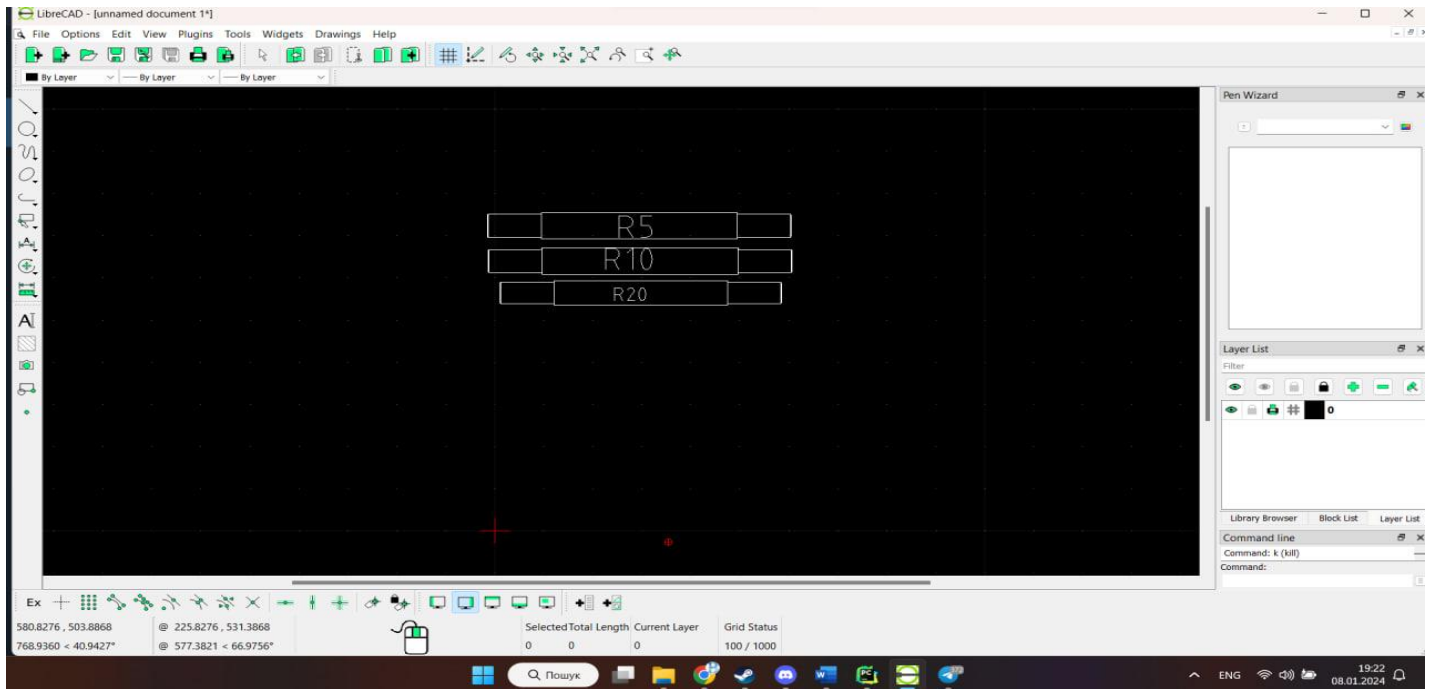


Рисунок 3.8 – Дослідження файлу ShaftR20

Далі переходимо до порівняння переважні ряди розмірів R5, R10, R20 та новий R40. Для цього нам потрібно знову запустити програму ввести ряд R40 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR40 після чого завантажимо цей файл у LibreCAD. Після чого ми отримали (рис. 3.9).

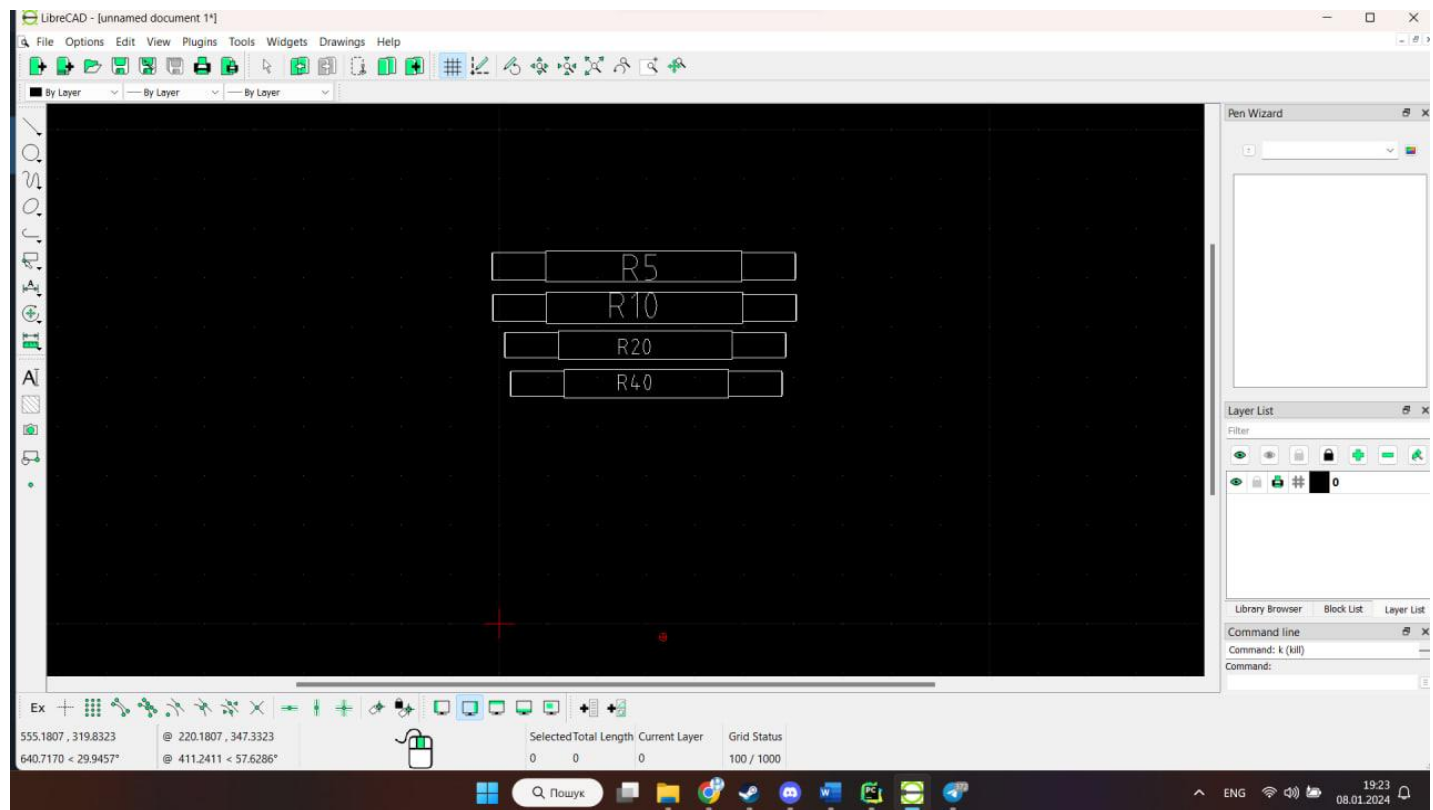


Рисунок 3.9 – Дослідження файлу ShaftR40

Після отриманого результату видно що отримали іншу деталь по розміру це сталося тому що переважні ряди розмірів R5, R10, R20 відрізняються від переважного ряду розмірів для R40 який має [1.00, 1.06, 1.12, 1.18, 1.25, 1.32, 1.40, 1.50, 1.60, 1.70, 1.80, 1.90, 2.00, 2.12, 2.24, 2.36, 2.50, 2.65, 2.80, 3.00, 3.15, 3.35, 3.55, 3.75, 4.00, 4.25, 4.50, 4.75, 5.00, 5.30, 5.60, 6.00, 6.30, 6.70, 7.10, 7.50, 8.00, 8.50, 9.00, 9.5, 10.00] такий ряд переважних розмірів. Тому переважний розмір діаметру 56 мм не змінився, а переважна довжина стала меншою 335 мм. Тому деталь стала меншою.

Далі переходимо до порівняння переважні ряди розмірів R5, R10, R20, R40 та новий R80. Для цього нам потрібно знову запустити програму ввести ряд R80 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR80 після чого завантажимо цей файл у LibreCAD. Після чого ми отримали (рис. 3.10).

Після отриманого результату видно що отримали іншу деталь по розміру це сталося тому що переважні ряди розмірів R5, R10, R20, R40 відрізняються від переважного ряду розмірів для R80 який має [1.00, 1.03, 1.06, 1.09, 1.12, 1.15, 1.18, 1.22, 1.25, 1.28, 1.32, 1.36, 1.40, 1.45, 1.50, 1.55, 1.60, 1.65, 1.70, 1.75, 1.80, 1.85, 1.90, 1.95, 2.00, 2.06, 2.12, 2.18, 2.24, 2.30, 2.36, 2.43, 2.50, 2.58, 2.65, 2.72, 2.80, 2.90, 3.00, 3.07, 3.15, 3.25, 3.35, 3.45, 3.55, 3.65, 3.75, 3.87, 4.00, 4.12, 4.25, 4.37, 4.50, 4.62, 4.75, 4.87, 5.00, 5.15, 5.30, 5.45, 5.60, 5.80, 6.00, 6.15, 6.30, 6.50, 6.70, 6.90, 7.10, 7.30, 7.50, 7.75, 8.00, 8.25, 8.50, 8.75, 9.00, 9.25, 9.50, 9.75, 10.00] такий ряд переважних розмірів. Тому переважний розмір діаметру 54.5 мм не змінився, а переважна довжина стала меншою 325 мм. Тому деталь стала меншою.

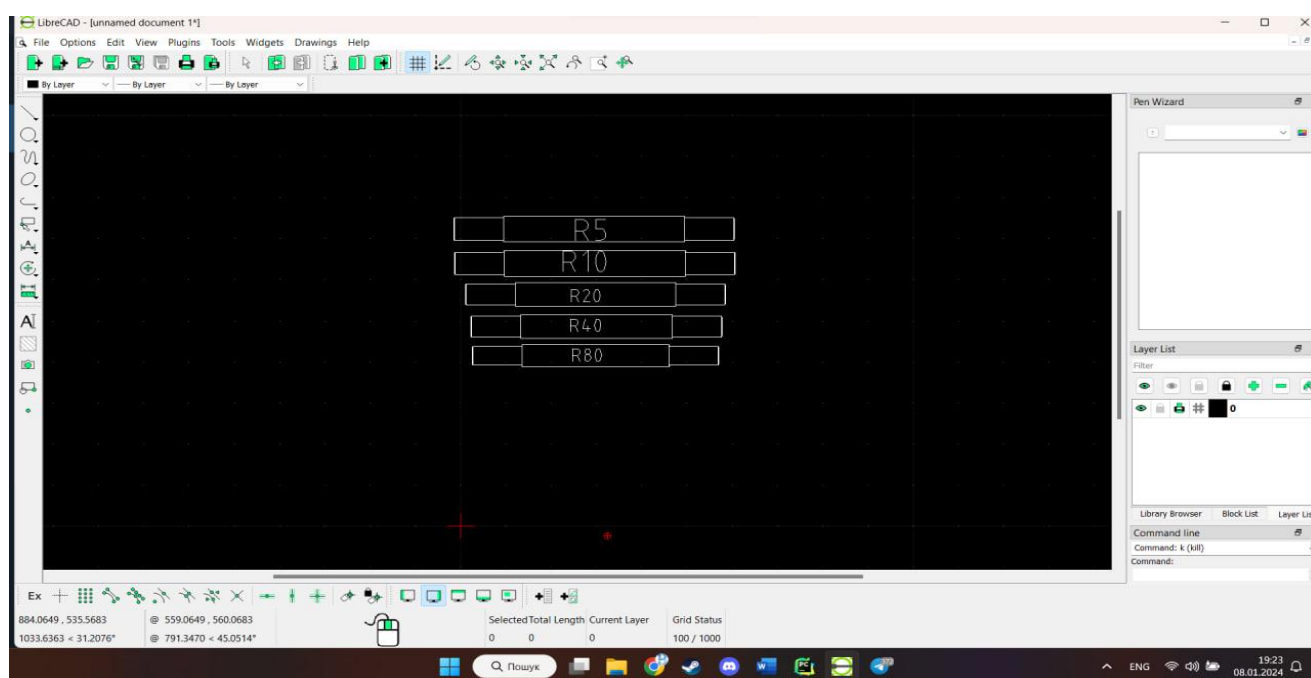


Рисунок 3.10 – Дослідження файлу ShaftR80

Далі переходимо до порівняння переважні ряди розмірів R5, R10, R20, R40, R80 та новий R160. Для цього нам потрібно знову запустити програму ввести ряд R160 та розмір діаметр 54,5 мм, а розмір довжини 317,5 мм та назвемо файл ShaftR160 після чого завантажимо цей файл у LibreCAD. Після чого ми отримали (рис. 3.11).

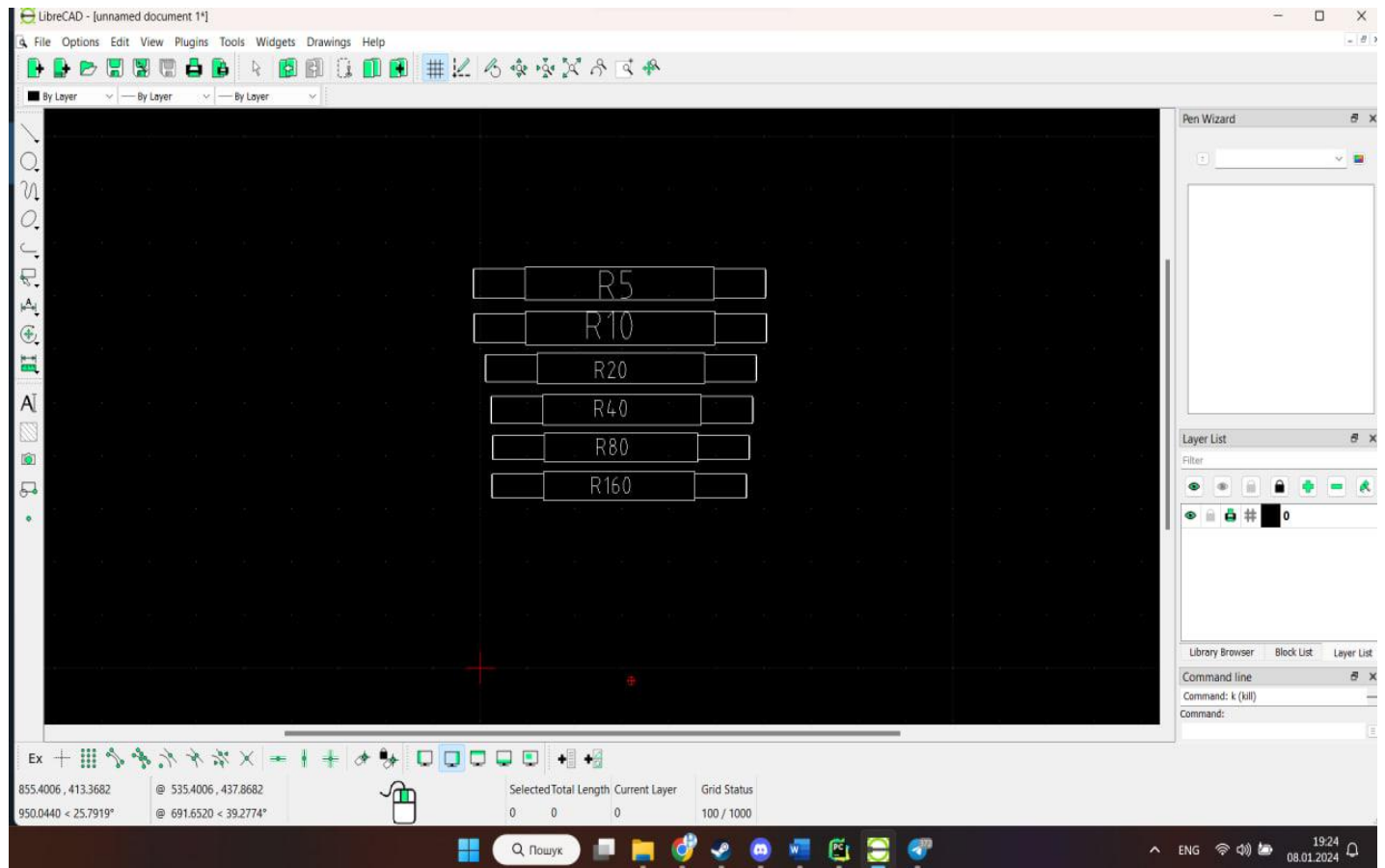


Рисунок 3.11 – Дослідження файлу ShaftR160

Після отриманого результату видно що отримали іншу деталь по розміру це сталося тому що переважні ряди розмірів R5, R10, R20, R40, R80 відрізняються від переважного ряди розмірів для R160 який має більший ряд за R80. Тому переважний розмір діаметру 54.5 мм не змінився, а переважна довжина стала меншою 320 мм. Тому деталь стала меншою але тільки по довжині.

Дослідивши залежність переважних розмірів для деталі було розглянуто п'ять варіантів рядів які впливають на деталь при цьому чим більший ряд тим більша ймовірність того що користувачу знайде той розмір який він би хотів який буде за ГОСТом.

### 3.3 Вплив стандартів проектування на об'єм проектованої циліндричної деталі

Стандарти проектування мають значний вплив на об'єм проектованої циліндричної деталі в інженерному дизайні.

Стандарти визначають типові розміри і толеранції для циліндричних деталей. Це сприяє уніфікації та стандартизації виробництва, дозволяючи використовувати стандартні деталі та матеріали, що полегшує виробничі процеси та підвищує сумісність з іншими компонентами.

Стандарти включають в себе вимоги до міцності та безпеки циліндричних деталей, що є важливим для їх ефективної експлуатації. Вони враховують фактори, такі як навантаження, температурні умови та інші експлуатаційні умови.

Сучасні стандарти також враховують вплив виробництва та експлуатації на навколишнє середовище. Це може включати вимоги до використання екологічно чистих матеріалів, обмеження викидів іншими забруднюючими речовинами та врахування можливості переробки чи утилізації.

Стандарти можуть також визначати економічно обґрунтовані підходи до проектування циліндричних деталей. Це може включати оптимізацію розмірів та матеріалів з метою зменшення витрат на виробництво та забезпечення ефективного використання ресурсів.

Стандарти визначають норми для взаємодії циліндричних деталей з іншими компонентами чи системами. Це важливо для забезпечення обманності та сумісності між різними виробниками та системами.

Для початку напишемо код за допомогою якого виведемо графік. Створений новий клас під назвою `Research_3`. В цьому класі ми відсилаємося на такі класи як `sizes`, `CylEndShafts`.

Створюємо функцію яка приймає п'ять параметрів:  $D$  (діаметр),  $L$  (довжина),  $x$  і  $y$  (координати початкової точки) та `file` (об'єкт файлу для запису команд). Вона обчислює координати трьох точок ( $p_1$ ,  $p_2$  і  $p_3$ ) на основі введених параметрів. Записує координати цих точок у файл. Генерує команди ліній (`li`) для з'єднання цих точок та закриваючу команду (`k`) для завершення малюнка. Ця функція фактично малює валик із зазначеним діаметром і довжиною, починаючи з вказаних координат ( $x$ ,  $y$ ), і записує відповідні команди LibreCAD у вказаний файл.

Створюємо функцію призначену для обчислення об'єму циліндричної деталі з отвором у валу за вказаними параметрами. Об'єм валу обчислюється як об'єм циліндра.

Додається об'єм від отвору у валу, обчислений як об'єм циліндра з відповідними параметрами. Додається об'єм від конусної частини, обчислений як об'єм конуса з відповідними параметрами. Використовуються `linspace` та `sqrt` для розбиття області на  $n$  точок і обчислення об'єму для кожної частини, який потім рахується та додається до загального об'єму. Отже, функція обчислює загальний об'єм циліндричної деталі з отвором у валу на основі заданих параметрів.

Наступна функція яка запитує користувача про введення діаметра та довжини валу, а потім обчислює та виводить відповідні розміри валу для кожного з вказаних стандартів (R5, R10, R20, R40, R80, R160). Далі для кожного розміру валу генерується файл із командами для LibreCAD, які включають малювання валу та кінцівок, а також визначається об'єм валу. Користувач вводить обчислений діаметр і довжину валу. Для кожного стандарту обчислюються бажані розміри валу та крайок (крайок отвору валу). Генерується файл із командами для LibreCAD для малювання валу та кінцівок. Обчислюється та додається об'єм валу у список  $V$ .

Робимо функцію яка виводить значення стандартів (`prefsizes`) та відповідні об'єми валів ( $V$ ) у списки, а потім будує стовпчикову діаграму за допомогою бібліотеки `matplotlib`. Виводяться значення стандартів та об'єми валів у вигляді списків. Використовується бібліотека `matplotlib` для побудови стовпчикової діаграми (`bar chart`). Діаграма має назву "Назва бажаних розмірів" для осі  $x$  та " $V, \mathrm{cm}^3$ " для осі  $y$ . Інструкція `ax.bar_label(p, label_type='edge')` додає мітки до стовпців діаграми. Виводиться діаграма. Завершується програма, очікуючи натискання клавіші `Enter`. Цей код буде виводити графік зі стовпчиками, де кожен стовпець відповідає одному стандарту, а висота стовпця відповідає об'єму валу для цього стандарту.

Дослідження всіх переважні розміри таких як R5, R10, R20, R40, R80, R160 та порівняємо як змінився об'єм на графіку. Об'єм зменшується кожен раз коли вибираємо більший переважний ряд. Це наглядно показано на рисунку (рис. 3.12).

Розглянув вплив на форму та об'єм кінцевого валу який буде більш кращим якщо ряд переважних чисел буде більшим.

Після детального дослідження всіх аспектів було зрозуміло що при збільшені переважних рядів розмірів було зменшено об'єм. При цьому досягнуто уніфікація та стандартизація розмірів, безпека та надійність, економічна ефективність, екологічні стандарти, сумісність та обмінності.

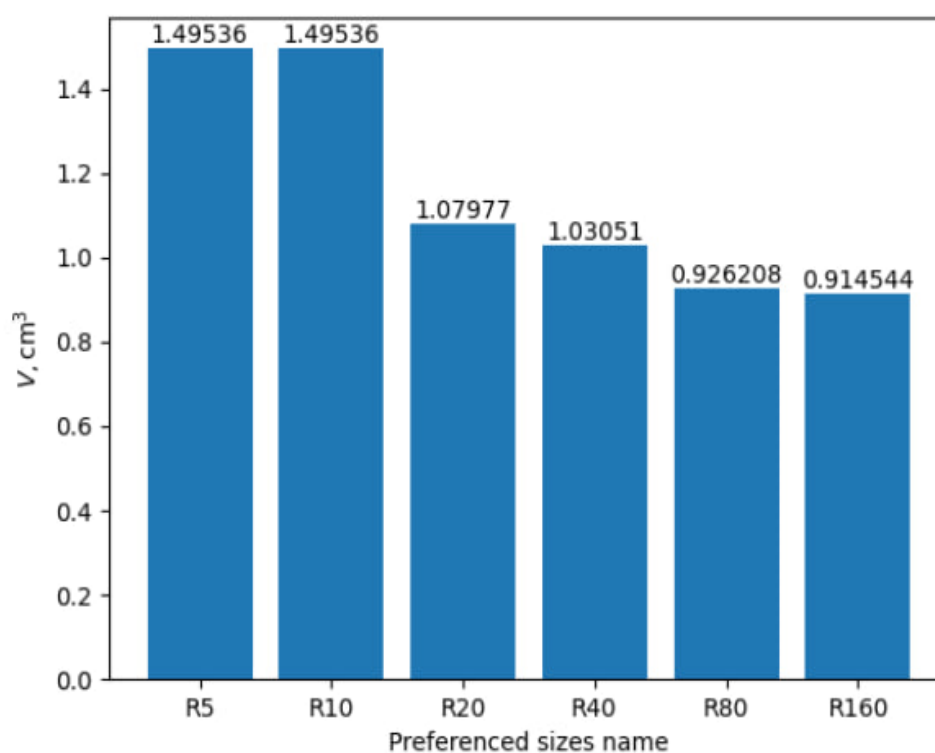


Рисунок 3.12 – Дослідження графіку впливу стандартів проектування на об’єм проєктованої циліндричної деталі

Стандарти визначають норми для взаємодії циліндричних деталей з іншими компонентами чи системами. Це важливо для забезпечення обманності та сумісності між різними виробниками та системами. Воно допомагає вирішити такі проблеми як:

- уніфікація та стандартизація розмірів;
- безпека та надійність;
- економічна ефективність;
- екологічні стандарти;
- сумісність та обмінності.

### 3.4 Виробнича безпека в лабораторії

Згідно до державної санітарної норми (ДСН) 3.3.6-042-99 робота в лабораторії з персональним комп’ютером (ПК) по категорії робіт відноситься до легких (категорія Ia – легкі фізичні роботи з енерговитратами до 139Вт або 120 ккал/г). Робота ведеться сидячи, не вимагає систематичного фізичного напруження і переміщення важких предметів.

У робочій зоні приміщення згідно з ДСН 3.3.6.042-99 повинні бути встановлені оптимальні значення параметрів мікроклімату. Для категорії робіт Іа в приміщенні повинні підтримуватися наступні параметри мікроклімату:

– температура повітря від 22 °С до 24 °С в теплий період і від 23 °С до 25 °С в холодний період;

– відносна вологість повинна бути в межах від 40 % до 60 %;

– швидкість руху повітря не повинна перевищувати 0,1 м/с.

В адміністративно-виробничих приміщеннях при організації робочих місць, оснащених персональними комп'ютерами, особливо велику увагу необхідно приділити освітленню. Освітлення при роботі з комп'ютером має свої риси, пов'язані з тим фактом, що зоровий аналізатор користувача комп'ютера при роботі отримує одночасно два світлових потоку: відбитий світловий потік світла від предметів що лежать поруч і прямого потоку світла від монітора комп'ютера.

Приміщення, в яких здійснюється експлуатація персональних комп'ютерів, обов'язково повинні мати одночасно і природне, і штучне освітлення, яке відповідає вимогам чинної нормативної документації.

Коефіцієнт природної освітленості в приміщеннях які експлуатуються із комп'ютерами не повинен бути нижче від 1,2 % до 1,5 %. Освітлення не повинно призводити до відблисків на поверхні екрану.

Освітленість поверхні екрану не може становити більше 350 лк. Слід обмежувати пряму блескність від джерел освітлення, при цьому яркі поверхні 62 (вікна, світильники та ін.), що знаходяться в полі зору, не може становити більше 250 кд/м<sup>2</sup>.

Показник осліпленості в разі використання загального штучного освітлення у виробничих та адміністративно-побутових приміщеннях повинен становити не більше 20.

Показник діскомфорту в адміністративно-громадських приміщеннях має бути не більше 40.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 °С до 90 °С з вертикаллю в подовжній і поперечній площинах обов'язково повинна бути менше 200 кд/м<sup>2</sup>, захисний кут світильників повинен становити більше 40 градусів.

Встановлені світильники місцевого освітлення повинні мати непрозорий відбивач із захисним кутом більше 40 градусів.

Згідно ДСН 3.36.037-99 необхідно забезпечити рівень шуму в лабораторії не більше 50 дБ. Забезпечення необхідного рівня шуму досягається використанням звукопоглинальних матеріалів.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проаналізовані та поставлені цілі які було виконано під час цієї роботи. Сучасне будівництво пристроїв постійно вдосконалюється, і однією зі сфер, які отримують значний науковий і практичний інтерес, є побудова деталей на підприємствах. Ця технологічна розробка має великий потенціал у різних сферах: промислове виробництво, архітектура та будівництво, авіаційна та автомобільна промисловість, машинобудування.

Для того щоби дослідити та зробити можливим роботу з автоматизованим проектуванням деталі. Було вибрана мова програмування система розробки та побудована блок схема та алгоритм програми. Після чого було написано код програми та досліджено всі впливи на деталь переважними розмірами.

Під час виконання кваліфікаційної роботи були досягнені такі цілі:

- проведено аналіз програм такі як AutoCAD,LibreCAD,Fusion360;
- проведено вибір потрібної програми;
- побудовано блок схему програми;
- вибрана мова програмування;
- проведений аналіз теоретичних даних;
- розглянуто систему проектування деталі в LiberCAD;
- написано код програми;

проаналізувати та дослідити вплив переважних розмірів на об'єм, розмір та вплив на деталь.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи [Електронний ресурс] – режим доступу: [https://tapr.nure.ua/wpcontent/uploads/2022/04/mv\\_\\_mag\\_151.pdf](https://tapr.nure.ua/wpcontent/uploads/2022/04/mv__mag_151.pdf)
2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
3. Білошапка І.В. Розробка методів щодо створення програмних модулів автоматизованого проектування деталей для системи LibreCAD [Електронний ресурс] / І.В. Білошапка, Ю.В.Ромашов// Автоматизація та Приладобудування збірник студентських наукових статей / Харківський національний університет радіоелектроніки; [редкол.: І.Ш.Невлюдов та ін.]. – Харків: ХНУРЕ, 2023. – Вип. 1. С.209-214 Режим доступу: <https://drive.google.com/file/d/1ZEр4lklhEkWoSUAZ5Pa6od2d12KQfdxe/view> (дата звернення: 15.10.2023).
- 4.AutoCAD. Моделювання веб-сайт. URL:<https://knowledge.autodesk.com>(дата звернення: 10.10.2023).
5. Що таке AutoCAD? Моделювання веб-сайт. URL: <https://uk.wikipedia.org/wiki/AutoCAD>(дата звернення: 12.10.2023).
6. Що таке Fusion360? Моделювання веб-сайт. URL:<https://uk.wikipedia.org/wiki/Fusion360>(дата звернення: 13.10.2023).
- 7.Fusion360. Моделювання веб-сайт. URL:<https://www.autodesk.com/products/fusion-360>(дата звернення: 13.10.2023).
- 8.Що таке LibreCad? Моделювання веб-сайт. URL: <https://uk.wikipedia.org/wiki/LibreCAD>(дата звернення: 15.10.2023).
- 9.Librecad. Моделювання веб-сайт. URL: <https://librecad.org/> (дата звернення: 17.10.2023).
10. Heikell J. LibreCAD for Real Dummies: навчальний посібник. Espoo, 2019. 97 p