

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ
РОЗРАХУНКУ ПАРАМЕТРІВ МЕХАНІЧНОЇ ОБРОБКИ ДЕТАЛЕЙ
RESEARCH AND SOFTWARE IMPLEMENTATION OF TOOLS TO
CALCULATE PARAMETERS OF
THE MECHANICAL PROCESSING OF PARTS

Виконав: студент 2 курсу, групи КНТ-122м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЗАКІНА К.Є.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЛПАКОВА Т.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЗАКІНОЇ Карини Євгеніївни

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів розрахунку параметрів механічної обробки деталей. Research and Software Implementation of Tools to Calculate Parameters of the Mechanical Processing of Parts

керівник проєкту (роботи) к.т.н., доцент, КОЛПАКОВА Тетяна Олексіївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 14 ” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЛПАКОВА Т.О., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 16 ” жовтня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Карина ЗАКІНА
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Тетяна КОЛПАКОВА
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
95 с., 35 рис., 3 табл., 3 дод., 17 джерел.

C++, C++BUILDER, МОВА ПРОГРАМУВАННЯ, ОБРОБКИ
ДЕТАЛЕЙ, ПРИПУСК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для
механічного оброблення деталей.

Предмет дослідження – програмні засоби для механічного оброблення
деталей.

Метою роботи є дослідження та розробка програмного забезпечення
для розрахунку параметрів механічної обробки деталей.

Матеріали, методи та технічні засоби: мова програмування C++,
середовище розробки C++Builder.

Результати. Розроблено програмне забезпечення для розрахунку
параметрів механічної обробки деталей за допомогою мови програмування
C++ та середовища розробки C++Builder.

Практична цінність роботи полягає у розробці програмного
забезпечення для розрахунку параметрів механічної обробки деталей, що
реалізує методи визначення припусків на обробку циліндричних та інших
типів поверхонь.

Висновки. Виконано проєктування програмного забезпечення для
розрахунку параметрів механічної обробки деталей. Розроблено програмне
забезпечення для розрахунку параметрів механічної обробки деталей.
Здійснено тестування розробленого програмного забезпечення для розрахунку
параметрів механічної обробки деталей.

Галузь використання – програмні засоби для автоматизації механічної
обробки деталей.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 95 pages, 35 figures, 3 tables, 3 appendixes, 17 sources.

C++, C++BUILDER, PROGRAMMING LANGUAGE, PARTS PROCESSING, FITTING, SOFTWARE.

The object of research is the process of developing software for mechanical processing of parts.

The subject of the research is software for mechanical processing of parts.

The purpose of this work is to develop the software for calculating the parameters of mechanical processing of parts.

Materials, methods and technical tools: C++ programming language, C++Builder development environment.

Results. The software tools for calculating the parameters of mechanical processing of parts using C++ programming language and C++Builder development environment have been developed.

The practical value of the work lies in the development of software for calculating the parameters of mechanical processing of parts, which implements methods for determining allowances for processing cylindrical and other types of surfaces.

Conclusions. The software for calculating the parameters of the mechanical processing of parts has been designed. The software for calculating the parameters of the mechanical processing of parts has been developed. The developed software for calculating the parameters of the mechanical processing of parts has been tested.

Scope of use – software tools for automating the mechanical processing of parts.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Поняття про механічну обробку та програмні засоби для автоматизації процесу обробки деталей машин та обладнання.....	11
1.2 Аналіз програмного забезпечення для підтримки процесу механічної обробки деталей	24
1.3 Висновки за розділом 1	33
2 Матеріали і методи	36
2.1 Вибір мови програмування	36
2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей .	38
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей	40
2.4 Висновки за розділом 2	43
3 Опис програми	44
3.1 Структура програмного забезпечення для розрахунку параметрів механічної обробки деталей	44
3.2 Функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей	46
3.3 Проєктування інтерфейсу програмного забезпечення для розрахунку параметрів механічної обробки деталей	48
3.4 Висновки за розділом 3	51
4 Експлуатація, тестування та експериментальне дослідження програми.....	53
4.1 Призначення й умови застосування програми	53
4.2 Характеристики програми для розрахунку параметрів механічної обробки деталей	54
4.3 Інструкція по експлуатації програми.....	54

4.3.1	Звернення до програми	54
4.3.2	Вхідні й вихідні дані	55
4.3.3	Повідомлення.....	55
4.4	Виконання програмного забезпечення для розрахунку параметрів механічної обробки деталей	55
4.5	Тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей	62
4.6	Висновки за розділом 4	63
	Висновки	64
	Перелік джерел посилання	67
	Додаток А Технічне завдання	69
	Додаток Б Фрагмент тексту програми	73
	Додаток В Слайди презентації	88

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CAD – Computer-Aided Design, комп’ютерна підтримка проєктування;

UML – Unified Modeling Language, уніфікована мова моделювання;

ООП – об’єктно-орієнтоване програмування;

ПЗ – програмне забезпечення;

ТЗ – технічне завдання;

ЧПК – числове програмне керування.

ВСТУП

У наш час при виготовленні деталей машин та обладнання широко використовуються програмні засоби. Традиційні методи розрахунків можуть бути часом невиправдано витратними, а використання програмних засобів дозволяє швидше та з меншими зусиллями отримувати результати. Програмні засоби дозволяють проводити розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективне використання матеріалів та інструментів [1]-[5].

Крім того, програмні рішення дозволяють здійснювати оптимізацію параметрів механічної обробки, забезпечуючи досягнення найкращих результатів при мінімальних витратах. Програмні засоби дозволяють виконувати аналіз різних варіантів обробки, враховуючи різні параметри, що дозволяє знайти найбільш оптимальний підхід. Деякі програмні засоби надають можливість моделювати процеси механічної обробки та виконувати симуляцію для передбачення результатів, підвищуючи таким чином ефективність процесу механічної обробки. Також використання програмних засобів допомагає уникнути людських помилок під час розрахунків та сприяє вищій точності результатів. Багато програмних засобів дозволяють створювати документацію та зберігати історію розрахунків, що полегшує відстеження та аналіз. Отже, використання програмного забезпечення для розрахунку параметрів механічної обробки деталей допомагає підвищити ефективність, точність та оптимізувати процеси виробництва в сучасній промисловості [1]-[5].

Проте, деякі програмні засоби можуть не забезпечити достатньої точності у розрахунках, особливо якщо моделі, на яких вони базуються, мають деякі недоліки або невідомі параметри. Для точних розрахунків потрібні якісні дані про матеріали, інструменти, параметри обробки тощо, а деякі програмні засоби використовують застарілі або спрощені моделі, які не враховують частину з цих параметрів. Інші програмні засоби можуть бути досить

складними у використанні, і вони вимагають від користувача високого рівня експертизи в області механічної обробки. Крім того, деякі програмні засоби для розрахунку параметрів механічної обробки деталей можуть бути платними або вимагати додаткових витрат на навчання персоналу [1]-[5]. Тому актуальною є розробка програмного забезпечення для розрахунку параметрів механічної обробки деталей.

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розрахунку параметрів механічної обробки деталей;
- розробити модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей;
- здійснити проєктування програмного забезпечення для розрахунку параметрів механічної обробки деталей;
- створити програмне забезпечення для розрахунку параметрів механічної обробки деталей;
- виконати тестування розробленого програмного забезпечення для розрахунку параметрів механічної обробки деталей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про механічну обробку та програмні засоби для автоматизації процесу обробки деталей машин та обладнання

Обробка деталей машин та обладнання є процесом створення бажаної форми та розміру матеріалу, зазвичай металу, шляхом здійснення контрольованого видалення матеріалу. Як правило, до цього процесу залучаються верстати та інше обладнання. Часто до цього процесу залучається спеціалізоване програмне забезпечення, що дозволяє керувати процесом обробки, а також розраховувати відповідні параметри виробничих процесів [1]-[6].

Механічна обробка є важливою складовою виробництва багатьох металевих виробів, але також може застосовуватися до інших матеріалів, таких як дерево, пластик, кераміка та композитні матеріали [1]-[7]. Більшість сучасних процесів механічної обробки виконуються з використанням числового програмного керування (ЧПК), де програмні засоби підтримують керування рухами і діями фрезерних верстатів, токарних верстатів та інших обробних машин. Це підвищує продуктивність, оскільки верстати з ЧПК можуть працювати без постійної присутності оператора, що зменшує витрати на робочу силу в машинних цехах [1]-[6].

Виробництво з використанням спеціалізованого програмного забезпечення включає процеси та операції, такі як процеси механічної обробки, процеси складання або процеси формування. Процеси механічної обробки, процеси складання, процеси формування необхідні для отримання кінцевого продукту із сировини. Кожен процес має свої переваги та недоліки, деякі процеси вимагають додаткової механічної обробки, наприклад гладкості поверхні, фінішна обробка тощо. це ті. Кожен тип процесу обробки полягає в отриманні точного та якісного кінцевого продукту [1]-[6].

Виготовлення потрібних деталей зі сировини пов'язано з обробкою різних матеріалів, які характеризуються різними властивостями, зокрема,

міцністю, твердістю, термостійкістю, довговічністю та іншими, які підлягають дослідженню. Ці властивості суттєво впливають на процес обробки [1]-[6].

Формування виробу може бути досягнуто шляхом різноманітних процесів, таких як різання, деформація, обробка поверхні та ін. [1]-[6].

Механічна обробка включає кілька видів процесів для завершення виробництва деталі. Процес механічної обробки вимагає додаткових дій, таких як центрифугування, точне лиття, лиття під тиском, лиття під тиском в холодній камері, різання, формування, фінішна обробка [1]-[6].

Завдяки сучасній технології виробництва з використанням спеціалізованого програмного забезпечення, механічна обробка використовується як для металів, так і для неметалів, включаючи важкі метали. Цей процес перетворює тверду сировину на необхідну форму, шляхом видалення надмірного матеріалу, що покриває деталь [1]-[6].

В процесі механічної обробки виконуються різноманітні операції, такі як свердління, фрезерування, різання, точіння, стругання, виконання канавок, притирання, протягування, шліфування тощо. Для цих завдань можуть бути застосовані різні інструменти, такі як дрилі, токарні верстати, розточувальні верстати, рубанки, гральні верстати, шліфувальні машини та інші [1]-[6]. Цей тип обробки став можливим завдяки прогресу в методах та технологіях виробництва [1]-[6].

Передові підходи до механічної обробки передбачають використання верстатів з ЧПК для досягнення вищої ефективності, покращеної якості продукції, скорочення часу виробництва та, найважливіше, забезпечення безпеки в різних галузях промисловості [1]-[6].

Сучасні технологічні інновації та програмне забезпечення значно змінили виробничі процеси та способи використання ріжучих інструментів на різних машинах. Шляхом поєднання високоякісних матеріалів ріжучих інструментів з різними видами сплавів, а також використанням спеціалізованого програмного забезпечення створюється дуже міцний і стійкий метал, що спрощує і полегшує процес роботи [1]-[6].

Виробництво може використовувати різноманітні технології, починаючи від ручних методів до стандартних механічних процесів та до більш високотехнологічних методів, таких як електрохімічна обробка, електронно-променева обробка, лазерне зварювання, ультразвукова обробка, обробка струменем води [1]-[6]. Велика кількість продукції, будь то побутові чи промислові товари, включаючи папір, меблі, пластик, метал та інші, проходить через процес механічної обробки [1]-[6].

Передова технологія виробництва, що спирається на використання спеціалізованого програмного забезпечення, сприяє не лише підвищенню швидкості виробництва, але й допомагає знизити кількість дефектів у продукції шляхом застосування різноманітних методів контролю, таких як контактна та безконтактна перевірка, руйнівне і неруйнівне тестування і т.д. Ці методи допомагають оцінити характеристики продукції та виявити можливі несправності якнайшвидше [1]-[6].

Процес збирання або складання механічних компонентів є способом об'єднання компонентів з метою отримання готового продукту. В даному контексті він включає незнімні з'єднання та кріплення [1]-[6]. Неперервні з'єднання є типом з'єднання, яке не передбачає роз'єднання, таке як зварювання чи паяння [1]-[6].

Зварний шов є міцним з'єднувальним методом, що застосовується для об'єднання як важких, так і легких металів. Зварний шов використовується виключно для металевих матеріалів. З'єднання, отримані шляхом зварювання, можуть функціонувати при високих температурах, витримувати значні навантаження та інші умови. У порівнянні, лужна пайка є іншим методом з'єднання, що застосовується для неметалевих матеріалів і не може витримувати високі температури [1]-[6]. Також існують кріпильні з'єднання, які не відносяться до категорії нероз'ємних з'єднань. Сюди входять різьбові з'єднання та заклепкові з'єднання. Різьбові кріплення, такі як гайки і болти, застосовуються в різьбових з'єднаннях. Їх можна зняти в будь-який момент, але заклепка не має бути так само легкою для зняття, як гайка. Якщо

видаляється прокладка, заклепка повинна ламатися [1]-[6]. Гайки та болти можуть бути використані багато разів, а заклепки тільки один раз [1]-[6].

Лиття є процесом формування, який використовує термічну обробку для створення необхідних матеріалів. Сировинний матеріал перетворюється на рідкий стан шляхом теплової обробки. У цьому процесі температура розплавленого матеріалу перевищує необхідну температуру для лиття перед тим, як він набере форму. Розплавлений матеріал потребує сталої температури, яка є необхідною для лиття [1]-[6]. Після того, як розплавлений матеріал заливається у форму, він починає процес затвердіння. По завершенні цього процесу готовий виріб виймається із форми [1]-[6]. Формування включає в себе різні методи, такі як лиття під тиском, пресування, видування та інші методи [1]-[6].

Механічна обробка в машинобудівній промисловості включає в себе виготовлення механічних деталей шляхом видалення матеріалу з металу або інших матеріалів за допомогою токарних верстатів з комп'ютерним числовим керуванням. Цей процес передбачає програмування технічних схем для обробки, підключення комп'ютера до верстата з ЧПК та виконання обробки згідно з програмованими інструкціями [1]-[6].

Перед початком механічної обробки важливо детально вивчити технічні креслення та характеристики деталей, зрозуміти виробничий процес та виміряти матеріал для відповідності вимогам, після чого з використанням спеціалізованого програмного забезпечення обрати відповідні параметри обробки. Під час обробки важливо виконувати самоконтроль та коригувати незбіги в статистичних даних. Після попередньої обробки і перед глибокою обробкою важливо перевірити вигляд і специфікації технічного креслення, а також точно виміряти контрольні точки [1]-[6]. У випадку масового виробництва деталей важливо спочатку визначити, чи відповідає зразок стандартам перед запуском виробництва [1]-[6].

Деталі піддаються обробці на основі поданих креслень або зразків замовників, використовуючи спеціалізоване програмне забезпечення та різні

види сировини, такі як нержавіюча сталь, мідні матеріали, алюмінієві сплави, залізні матеріали та інші. Для обробки використовуються пристрої, такі як верстати з ЧПК, токарні та фрезерні верстати, дрилі, полірувальні машини тощо. Цей процес застосовується для створення різноманітних деталей, таких як гвинти, вали двигуна, деталі моделей автомобілів, інструменти для лову, коробки динаміків, блоки живлення для мобільних пристроїв [1]-[6].

Типові методи виготовлення металевих деталей включають механічну обробку, штампування, точне лиття, порошкову металургію та лиття металу під тиском. Металева фурнітура включає в себе різноманітні компоненти та пристосування, які використовуються в будівництві або меблевій промисловості, наприклад, цвяхи, шурупи, дроти, замки, петлі, засуви, пружини та інші. Метал є важливим матеріалом для промисловості, а його види включають золото, срібло, мідь, залізо та олово. Метал відіграє ключову роль у розвитку промислового виробництва та відіграє важливу роль в забезпеченні національного прогресу [1]-[6].

Завдяки широкому асортименту продукції та різноманітним характеристикам, металева фурнітура виконує важливу роль у різній діяльності. Вибір правильних металевих аксесуарів може зробити різні декоративні матеріали більш безпечними та зручними для використання. На сьогоднішній день на ринку доступно понад десять різних типів металевої продукції [1]-[6].

Існує множина видів металевої продукції з різними технічними характеристиками, і вони також відіграють важливу роль у декоративному оформленні будівель [1]-[6].

Сучасні методи обробки матеріалів, такі як точіння, фрезерування, стругання, шліфування, розточування тощо, постійно розвиваються та використовують спеціалізоване програмне забезпечення, включаючи і обробку за допомогою електроерозійних процесів. Крім цього, до засобів обробки входить лиття під тиском та кування [1]-[6].

Обробку металу можна класифікувати на різні види: автоматичне токарне оброблення, обробку з використанням числового програмування, токарну обробку на верстатах з ЧПК, а також обробку на п'ятиосьових токарних верстатах. Зазвичай її поділяють на обробку поверхні металу та обробку металу під впливом тиску [1]-[6].

Швидке створення прототипів з використанням програмних засобів часто включає в себе використання обробки з числовим керуванням, розробку дизайну на кресленнях, програмування ЧПК, обробку, шліфування, збирання, заливання, окислення, лазерне гравірування та інші етапи [1]-[6].

Обробка деталей за допомогою верстатів з ЧПК пов'язано з виконанням інструкцій спеціального програмного забезпечення. Програма створює маршрут та видаляє матеріал з алюмінієвого сплаву, створюючи бажаний прототип [1]-[6].

Після обробки на верстаті з ЧПК можуть залишитися задирки та сліди. Ці машини не завжди регулюються належним чином, тому потрібне ручне втручання. Зазвичай, для полірування використовують полірувальну пасту або проводять шліфування. Під час процесу полірування можуть виникнути певні проблеми. Наприклад, використання грубої шкурки для первинної обробки та шліфування, щоб видалити великі недоліки, а потім дрібної для відшліфування [1]-[6].

При розробці програмного забезпечення для розрахунку параметрів механічної обробки деталей необхідно враховувати використання та технічні характеристики виробу, спрощуючи процес монтажу та можливості ремонту. Також необхідно сприяти оптимізації використання металевих матеріалів, мінімізуючи різноманітність матеріалів та специфікацій для зменшення матеріальних витрат. Програмне забезпечення для підтримки виробничих процесів також повинно забезпечувати зручність виготовлення великої кількості процесів та сприяння готовності до механізованого та автоматизованого виробництва для підвищення продуктивності праці [1]-[6].

Проаналізуємо етапи оброблення деталей з використанням спеціалізованого програмного забезпечення. На першому етапі відбувається вибір матеріалу заготовки. У відповідності до проекту виготовлення деталей, вибір заготовки полягає в визначенні розмірів заготовки після врахування типу матеріалу, технологічних процесів та допусків [1]-[6].

На наступному етапі визначаються опорні точки позиціонування для складних частин. Під час вибору точок, на яких деталі будуть закріплюватись під час обробки, важливо врахувати відповідність розмірів та положення між необробленою та обробленою поверхнею, що відповідає вимогам креслення. Це забезпечує наявність достатніх запасів для обробки кожної поверхні. Можливо вибрати не повністю оброблену поверхню як площину позиціонування [1]-[6].

Далі виконується розробка шляху обробки деталей. Основною метою цього етапу є забезпечення належного зазору на обробленій поверхні. Залежно від способу обробки деталей та отворів, визначається позиція опори, яка використовується для позиціонування [1]-[6].

Виробничий процес обробки деталей з використанням спеціалізованого програмного забезпечення охоплює послідовність етапів або дій, пов'язаних з перетворенням сировини або готових компонентів на кінцеві продукти або компоненти. Від планування та проектування до виробництва та збирання, цей процес охоплює різноманітні дії з метою створення готової продукції, що відповідає конкретним вимогам. Виробничі процеси поділяються за їх призначенням, такі як лиття, кування, механічна обробка, зварювання, з'єднання та інші. Вибір конкретного виробничого процесу залежить від різних факторів, таких як необхідні допуски, обсяги виробництва та витрати. Важливо постійно вдосконалювати виробничі процеси для досягнення якісних, ефективних та економічно вигідних результатів [1]-[6].

Проаналізуємо різні види виробничих процесів. Повторне виробництво полягає у масовому виготовленні одного типу продукції великими кількостями. Дискретне виробництво направлено на створення унікальних

виробів та матеріалів із специфічними вимогами та характеристиками. Цехове виробництво займається виготовленням обмеженої кількості продукції на замовлення, відповідно до замовника. Безперервне виробництво використовується у хімічній або нафтовій промисловості, де виробничий процес не зупиняється. Серійне виробництво займається виробництвом продукції в партіях заздалегідь визначеної кількості, яке характерне для харчової та фармацевтичної галузей [1]-[6].

Повторне виробництво є формою виготовлення, при якій створюються значні кількості ідентичних або схожих продуктів. Цей вид виробництва переважно використовується для товарів, що мають високий попит і піддаються стандартизованим виробничим процесам, наприклад, пляшкової води. Повторний виробничий процес спрямований на мінімізацію часу і зусиль, потрібних для виготовлення кожної одиниці, та забезпечення високої якості всіх виробів [1]-[6].

Для досягнення цього зазвичай використовують спеціалізоване програмне забезпечення та обладнання, спроектоване для виконання конкретних завдань, крім того виконують постійні калібрування між процесами. Повторні виробничі процеси можуть бути вкрай ефективними і прибутковими, дозволяючи виробникам економити на масштабі та зменшувати витрати на працю, матеріали та обладнання [1]-[6]. Однак вони можуть також бути обмеженими у гнучкості, оскільки машини та обладнання розроблені для конкретних завдань і не піддаються легкій перепроєктуванню для виробництва інших продуктів [1]-[6].

Дискретне виробництво відображає підхід, де формуються індивідуальні продукти, що мають чітку ідентифікацію і часто виробляються в невеликих кількостях. Цей процес може включати складні матеріали, що вимагають кількох етапів обробки, використання різних матеріалів і обладнання, а також спеціалізованого програмного забезпечення [1]-[6]. Дискретне виробництво застосовується в таких галузях як електроніка, автомобільна промисловість, техніка, меблі, іграшки та багато інших. Воно

становить основний метод виробництва, особливо популярний в автомобільній промисловості [1]-[6]. Дискретне виробництво вимагає високої точності через індивідуальний підхід до кожного продукту, тому для його підтримки широко використовуються спеціальні програмні засоби. Навіть невеликі помилки можуть негативно вплинути на якість кінцевого продукту [1]-[6].

Цехове виробництво передбачає створення індивідуальних або спеціалізованих виробів, які часто мають унікальні характеристики та вимагають особливих навичок та процесів для їх виготовлення. Ці продукти зазвичай призначені для відповіді на конкретні потреби та зазвичай створюються на замовлення або індивідуально [1]-[6]. Цехове виробництво активно застосовується у сферах аерокосмічної, автомобільної та медичної промисловості, де попит на унікальне спеціалізоване обладнання є високим. Основною відмінністю є гнучкість та можливість виготовлення різноманітних продуктів, для чого часто використовується спеціалізоване програмне забезпечення [1]-[6].

Цей процес часто включає комбінацію ручних та автоматизованих методів та може потребувати спеціалізованих програмних засобів та інструментів, обладнання та кваліфікованого персоналу. Кожне завдання чи замовлення має свої унікальні характеристики, і процес виготовлення повинен бути адаптований під конкретні вимоги продукту. Час виробництва може бути значно більшим у порівнянні з іншими методами виробництва. Термін виконання зазвичай визначає період між завершенням виробництва та доставкою продукту замовнику [1]-[6].

Серійне виробництво відноситься до типу виробничого процесу, де виробництво продукту відбувається у визначеній кількості за один раз. У цьому процесі ресурси, такі як сировина, обладнання та робоча сила, використовуються протягом всього циклу партії [1]-[6]. Зазвичай пакетні процеси застосовуються у виробництві фармацевтичних препаратів та інших товарів, які вимагають специфічних рецептур або мають обмежений термін

зберігання. Пакетний процес включає кілька етапів, включаючи підготовку партії, обробку та обробку. На етапі підготовки партії здійснюється збір необхідної сировини та налаштування обладнання для виробництва, зокрема, за допомогою спеціалізованого програмного забезпечення [1]-[6].

Для забезпечення неперервності виробничого процесу всі ресурси, включаючи сировину, обладнання та робочу силу, використовуються для стабільного виробництва. Безперервні процеси зазвичай використовуються у виробництві хімічних речовин, добрив, електроенергії та паперу. У цих випадках сировина постійно зберігається в устаткуванні та перетворюється на продукт, поки він не буде готовий для упаковки чи розподілу. Такий підхід означає, що процес не переривається протягом виробничого циклу та не впливає на постійне виробництво [1]-[6]. Однією з переваг неперервного виробничого процесу є його висока ефективність, оскільки для отримання великого обсягу продукції потрібно менше часу та зусиль. Це також допомагає досягнути більшої стабільності та однорідності кінцевого результату [1]-[6]. Однак варто зазначити, що неперервний процес може бути менш гнучким, оскільки важко внести зміни в виробничий процес чи рецептуру продукту після його запуску [1]-[6].

Механічна обробка є ключовим етапом виробництва в галузі машинобудування. Це контрольований процес, під час якого матеріал (зазвичай металевий) знімається, щоб створити компоненти у бажаній формі та розмірі. Цей метод широко використовується для створення різних металевих виробів, а також може бути застосований до інших матеріалів, таких як дерево, пластик, кераміка та композити [1]-[6]. Відмінність такого виробництва від адитивного виробництва полягає в тому, що матеріали видаляються, а не додаються [1]-[6].

Адитивне виробництво вказує на метод виготовлення тривимірних об'єктів шляхом послідовного додавання тонких шарів розплавленого або частково розплавленого матеріалу. У цьому процесі кожен новий шар надійно зливається з попереднім. Програмне забезпечення для автоматизованого

проектування використовує цифрові моделі для розрізання на тонкі шари та генерації відповідних даних [1]-[6].

Терміни "3D-друк" та "адитивне виробництво" часто використовуються як взаємозамінні. Адитивне виробництво є більш широким поняттям, порівняно з 3D-друком, оскільки охоплює різноманітні процеси. Приклади адитивного виробництва включають виробництво за допомогою електронного променя та селективного лазерного плавлення. В обох випадках використовується послідовне додавання матеріалу для створення об'єктів, але жоден з них не передбачає виготовлення продукту шарами [1]-[6].

Процес 3D-друку відомий як одна з форм адитивного виробництва, де тривимірні об'єкти формуються шляхом поступового нанесення шарів матеріалу [1]-[6]. Для здійснення цього використовується спеціальний 3D-принтер, який перетворює цифрову модель або проект фізичного об'єкту [1]-[6]. Процес 3D-друку розпочинається зі створення цифрової моделі об'єкта за допомогою програмного забезпечення автоматизованого проектування або шляхом використання технології 3D-сканування для отримання цифрового зображення [1]-[6]. Після цього ця цифрова модель передається в 3D-принтер, який поступово створює об'єкт, додаючи матеріал шар за шаром. В якості матеріалів можуть використовуватися пластик, метал, сплави, полімери, і навіть біологічні матеріали, такі як тканини [1]-[6].

Однією з ключових переваг 3D-друку є можливість швидко і зручно створювати складні та індивідуальні об'єкти. Це також ефективніше з точки зору витрат, порівняно з традиційними методами виробництва для виготовлення невеликих серій виробів. 3D-друк допомагає зменшити відходи, оскільки використовується лише той матеріал, який потрібен для створення об'єкта. Ця технологія застосовується в різних галузях, таких як авіаційна, автомобільна, медична та виробництво споживчих товарів, для створення прототипів, форм та кінцевих продуктів [1]-[6].

Управління різноманітними видами виробничих процесів за допомогою програмного забезпечення вимагає нагляду за створенням товарів або послуг з

використанням різних засобів, ресурсів та персоналу [1]-[6]. Для успішного управління виробничими процесами за допомогою програмного забезпечення важливо враховувати кілька ключових аспектів [1]-[6]:

- планування включає розробку стратегій досягнення виробничих цілей, визначення необхідних ресурсів та розподіл завдань для забезпечення своєчасної виробництва;

- контроль якості: забезпечення відповідності продукції встановленим стандартам якості є важливим аспектом виробництва. Це можна здійснити через перевірки, тестування та аудит;

- управління запасами: оптимальний баланс між виробничими потребами та наявними ресурсами суттєво впливає на безперебійну роботу виробничих процесів;

- планування виробництва: включає концепцію вчасного виробництва, коли матеріали постачаються на виробничі лінії саме в потрібний момент;

- постійне вдосконалення, що передбачає систематичне спостереження за виробничими процесами та вдосконалення продуктивності шляхом виявлення неефективних процесів, впровадження покращень та нових технологій [1]-[6].

Зосереджуючись на цих ключових аспектах та використанні спеціалізованого програмного забезпечення можна підвищити ефективність, знизити витрати та підвищити загальну продуктивність виробничого процесу [1]-[6].

Важливо відзначити, що при механічній обробці деталей машин та обладнання початкове креслення деталі відрізняється від готового креслення деталі тим, що на всіх оброблених поверхнях передбачені додаткові припуски для подальшої обробки. У галузі машинобудування в процесі обробки матеріалу використовують різні види припусків: загальний припуск на перетворення, робочий припуск, який враховується під час конкретної операції, та проміжний припуск, який застосовується для переходу між операціями [1]-[6].

Розмір заданого припуску визначається в залежності від способу виготовлення деталі за обраною технологією, товщини шару дефектів та стану використовуваного обладнання. Під час обробки деталі фактичні розміри для експлуатації, як правило, не можуть бути досягнуті з абсолютною точністю, і тому реальні значення припусків коливаються в певних межах. В цьому контексті виділяють мінімальні, номінальні (розрахункові) та максимальні значення припусків [1]-[6].

Точний розрахунок та вибір припусків та робочих припусків під час механічної обробки деталей є ключовим для розробки технологічних процесів, від яких залежать вартість, якість, тривалість експлуатації та надійність. Розмір припусків впливає на економічність процесу виробництва. Більший припуск вимагає більше проходів для видалення відповідного металевого шару, збільшуючи трудомісткість, витрати енергії, різання інструменту та витрати на утворення стружки, а також потребує більше обладнання для виробництва. Зменшення припуску не завжди гарантує видалення дефектів поверхні металу для досягнення потрібної точності та шорсткості, і може навіть спричинити негативні наслідки [1]-[6]. Тому актуальною є розробка та впровадження програмних засобів для розрахунку параметрів механічної обробки деталей, які дозволять автоматизувати відповідні процеси та здійснювати розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективне використання матеріалів та інструментів.

Визначено, що у наш час при виготовленні деталей машин та обладнання широко використовуються програмні засоби, оскільки програмні засоби дозволяють проводити розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективну використання матеріалів та інструментів. Проаналізовано процес механічної обробки, що є важливою складовою виробництва багатьох виробів, визначено, що більшість сучасних процесів механічної обробки виконуються з використанням програмних засобів, які підтримують керування рухами і діями фрезерних верстатів. Проаналізовано процеси та операції виробництва та механічної обробки з

використанням спеціалізованого програмного забезпечення. Проаналізовано різні види виробничих процесів, а також етапи оброблення деталей з використанням спеціалізованого програмного забезпечення. Відзначено, що при механічній обробці деталей машин та обладнання початкове креслення деталі відрізняється від готового креслення деталі тим, що на всіх оброблених поверхнях передбачені додаткові припуски для подальшої обробки. Точний розрахунок та вибір припусків та робочих припусків під час механічної обробки деталей є ключовим для розробки технологічних процесів, від яких залежать вартість, якість, тривалість експлуатації та надійність, що обґрунтовує доцільність розробки програмних засобів для розрахунку параметрів механічної обробки деталей, які дозволять автоматизувати відповідні процеси та здійснювати розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективне використання матеріалів та інструментів.

1.2 Аналіз програмного забезпечення для підтримки процесу механічної обробки деталей

Проаналізуємо програмне забезпечення для підтримки процесу механічної обробки деталей [7]-[13].

Програмне забезпечення Fusion 360 представляє собою програмне забезпечення для підтримки виробничого процесу, комп'ютерного проєктування деталей, проєктування обладнання за доступною ціною. Це програмне рішення може бути ефективним засобом для невеликих підприємств або для проєктування нескладних деталей [10], [11].

Однією з важливих переваг Fusion 360 (рис. 1.1) є можливість безпосереднього моделювання. В порівнянні, наприклад, з SolidWorks, що є досить дорогим програмним забезпеченням, можна зазначити, що функції безпосереднього моделювання в програмному забезпеченні Fusion 360 працюють досить ефективно [10], [11].

 Fusion 360

[Contact sales at +91 8010340891](#)

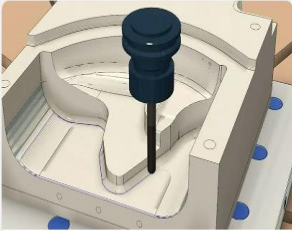
[Download free trial](#)

[See Fusion 360 pricing](#)

[Fusion 360 Overview](#)
[By role](#)
[Extensions](#)
[Start free trial](#)
[Buy Fusion 360](#)

Fusion 360 extensions

Add extensions with flexible subscription options to fit your project budget.

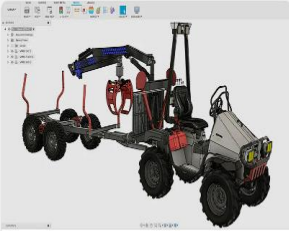


Machining Extension

Advanced manufacturing tools, include 3 to 5-axis strategies, toolpath optimisation and process automation.

[Download free trial](#)

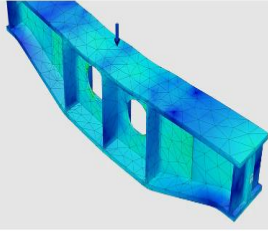
₹95,580/year



Nesting & Fabrication Extension

Create optimised and associative multi-sheet layouts for sheet metal and non-sheet metal parts automation.

[Download free trial](#)



Simulation Extension

Get unlimited access to cloud solves. Use structural, thermal, explicit and injection moulding simulations.

[Download free trial](#)

₹95,580/year



Generative Design Extension

Explore unlimited generative designs. Optimise for specific materials and manufacturing techniques.

[Download free trial](#)

Рисунок 1.1 – Програмне забезпечення Fusion 360 [10], [11]

Програмне забезпечення Fusion 360 також демонструє високу якість візуалізації та може навіть відображати рух механізмів. Безкоштовна можливість використання вдосконаленого вбудованого модулю становить додаткову перевагу, особливо для студентів або особистого використання. З цими функціями та доступністю я очікую, що популярність Fusion 360 буде зростати в майбутньому [10], [11].

Проте важливо відзначити, що Fusion 360 не є ідеальним вибором для виконання складного механічного аналізу. В даний час обмежена кількість невеликих компаній використовує Fusion 360, але можливо, у майбутньому більше невеликих підприємств перейдуть на це рішення [10], [11].

З щорічною вартістю приблизно 545 доларів для комерційного використання, Fusion 360 відзначається як досить доступне програмне забезпечення для комп'ютерного проектування інженерних рішень [10], [11].

Серед особливостей програмного забезпечення Fusion 360 для підтримки процесу механічної обробки деталей можна виділити такі [10], [11]:

- широкий функціонал: програмне забезпечення Fusion 360 надає можливість для комп'ютерного проектування, моделювання, створення креслень, а також використання вбудованих інструментів для механічного оброблення та аналізу [10], [11];

- безпосереднє моделювання: на відміну від інших програм, програмне забезпечення Fusion 360 має потужні інструменти для безпосереднього моделювання, що спрощує процес творення та редагування складних форм [10], [11];

- засоби інтеграції: програмне забезпечення Fusion 360 має вбудований комп'ютерний аспект виробництва, що дозволяє створювати і оптимізувати виробничі процеси для механічної обробки [10], [11];

- візуалізація механізмів: програмне забезпечення Fusion 360 може демонструвати рух механізмів, що допомагає уявити і перевірити їх функціонування перед фактичним виготовленням [10], [11];

- підтримка для металевих та інших матеріалів: програмне забезпечення Fusion 360 спеціально розроблено для механічної обробки металевих виробів, а також може застосовуватися до інших матеріалів, таких як дерево, пластик, кераміка тощо [10], [11];

- хмарна інтеграція: програмне забезпечення Fusion 360 пропонує можливість зберігання та синхронізації проектів у хмарних платформах, що дозволяє доступ до проектів з різних пристроїв та спільну роботу над ними з командою [10], [11];

- можливість безкоштовного використання: для студентів та навчальних закладів Fusion 360 надає безкоштовну ліцензію, що сприяє поширенню та вивченню програми [10], [11];

- спрощений процес виробництва: завдяки вбудованим програмним модулям та іншим функціям Fusion 360 може зменшити час та зусилля, необхідні для підготовки деталей до виробництва [10], [11];

– засоби спільної роботи над проектами: завдяки хмарній інтеграції та спеціальним інструментам, програмне забезпечення Fusion 360 сприяє спільній роботі в команді над проектами, спрощуючи обмін інформацією та взаємодію [10], [11];

– широкий спектр інструментів для аналізу: програмне забезпечення Fusion 360 дозволяє проводити різноманітний аналіз, включаючи аналіз напружень, теплопередачі, руху механізмів та інші, що сприяє вдосконаленню проектів перед виробництвом [10], [11];

– моделювання для 3D-друку: програмне забезпечення Fusion 360 має інтеграцію з 3D-друку, дозволяючи моделювати та перевіряти деталі під кутом, враховуючи особливості цього виробничого процесу [10], [11];

– можливості для інновацій: програмне забезпечення Fusion 360 підтримує різноманітні додатки та розширення, що дає можливість розробникам створювати інноваційні рішення та додатки для програми [10], [11];

– підтримка багатьох платформ: програмне забезпечення Fusion 360 доступний для Windows, macOS та операційні системи для мобільних пристроїв, що дозволяє працювати з проектами в різних середовищах.

Серед недоліків програмного забезпечення Fusion 360 можна відзначити те, що для нових користувачів інтерфейс Fusion 360 може здатися складним і заплутаним через велику кількість функцій та інструментів. Крім того, під час роботи з великими і складними проектами можуть виникати обмеження продуктивності програми. Програмне забезпечення Fusion 360 може вимагати відносно потужного обладнання для оптимальної продуктивності, що може бути проблемою для користувачів зі старими або менш потужними комп'ютерами [10], [11].

Програмне забезпечення Abaqus представляє собою програмне забезпечення для автоматизації підтримки процесу механічної обробки деталей за допомогою методу аналізу кінцевих елементів [10], [12].

Програмне забезпечення Abaqus широко використовується у багатьох галузях промисловості. Професійні інженери, які працюють в області виробництва, високо оцінюють Abaqus за його мультифізичні можливості, що дозволяють їм докладно вивчати виробничі компоненти [10], [12].

Одним з ключових аспектів Abaqus (рис. 1.2) є можливість прямого моделювання, що дозволяє створювати та редагувати моделі безпосередньо. Це важливо для ефективної роботи з виробничими деталями [10], [12].



The ABAQUS Learning Edition is available free of charge to anyone wishing to get started with Abaqus.

The Abaqus Learning Edition is available on Windows platform only and supports structural models up to 1000 nodes. The full documentation collection in HTML format makes this the perfect Abaqus learning tool. You can download the ABAQUS Learning Edition free of charge from the

Рисунок 1.2 – Програмне забезпечення Abaqus [10], [12]

Крім цього, програмне забезпечення Abaqus надає широкий вибір матеріалів та можливість моделювати нелінійні матеріали. Воно може розв'язувати задачі нелінійності та надавати явні рішення. Програма Abaqus використовується у різних галузях, таких як автомобільна та аерокосмічна промисловість. Програмне забезпечення Abaqus є популярною серед професійних інженерних організацій завдяки можливостям моделювання матеріалів і налаштуванню програми [10], [12]. Незважаючи на це, варто

відзначити, що Abaqus може бути не найкращим вибором для розширеного механічного аналізу [10], [12].

Програмне забезпечення Abaqus у порівнянні з іншими програмами для підтримки процесу механічної обробки деталей, є потужним інструментом з широким функціоналом [10], [12].

Серед особливостей програмного забезпечення Abaqus для підтримки процесу механічної обробки деталей можна виділити такі [10], [12]:

- засоби інтегрованого середовища розробки: програмне забезпечення Abaqus надає інтегроване середовище для розробки моделей, встановлення умов, запуску симуляцій та аналізу результатів [10], [12];

- засоби фізичного аналізу: програма Abaqus підтримує фізичний аналіз, що дозволяє враховувати взаємодію різних фізичних явищ, таких як механіка, тепло, електромагнетизм, акустика тощо. Це особливо корисно для реалістичного моделювання різних умов [10], [12];

- моделювання нелінійних матеріалів: програмне забезпечення Abaqus надає можливість моделювати поведінку нелінійних матеріалів, таких як пластичні та інші, що дозволяє враховувати реалістичні властивості матеріалів [10], [12];

- засоби прямого моделювання: програма Abaqus дозволяє створювати та редагувати моделі безпосередньо, що спрощує процес розробки та моделювання [10], [12];

- моделювання контакту та взаємодії: програмне забезпечення Abaqus дозволяє моделювати складні взаємодії між різними деталями та поверхнями, включаючи стикання, розділення, ковзання та інші види контакту [10], [12];

- засоби динамічного аналізу: програмне забезпечення Abaqus підтримує аналіз динамічних процесів, включаючи вібрації, удари та інші динамічні навантаження [10], [12];

- широкий вибір матеріалів: програмне забезпечення Abaqus має вбудовану базу даних з різноманітними матеріалами, а також дозволяє користувачам створювати власні моделі матеріалів [10], [12];

– моделювання елементів зв'язку: програмне забезпечення Abaqus дозволяє створювати елементи зв'язку, що дозволяють моделювати поведінку з'єднань, зварних швів та інших з'єднань [10], [12].

Серед недоліків програмного забезпечення Abaqus можна відзначити те, що для нових користувачів інтерфейс Abaqus може здатися складним і заплутаним через велику кількість функцій та інструментів. Крім того, складність моделювання та аналізу у програмі Abaqus може призводити до великих витрат часу на підготовку, налаштування та виконання моделювання. Програмне забезпечення Abaqus може вимагати відносно потужного обладнання для оптимальної продуктивності, що може бути проблемою для користувачів зі старими або менш потужними комп'ютерами [10], [12].

Програмне забезпечення Caesar II (рис. 1.3) є програмним забезпеченням для автоматизації підтримки процесу механічної обробки деталей та аналізу напружень в трубопроводах у сфері машинобудування. Програмне забезпечення Caesar II має широкий спектр можливостей для професійного дослідження, виготовлення та проектування трубопроводів, а також відповідає галузевим стандартам [10], [13].

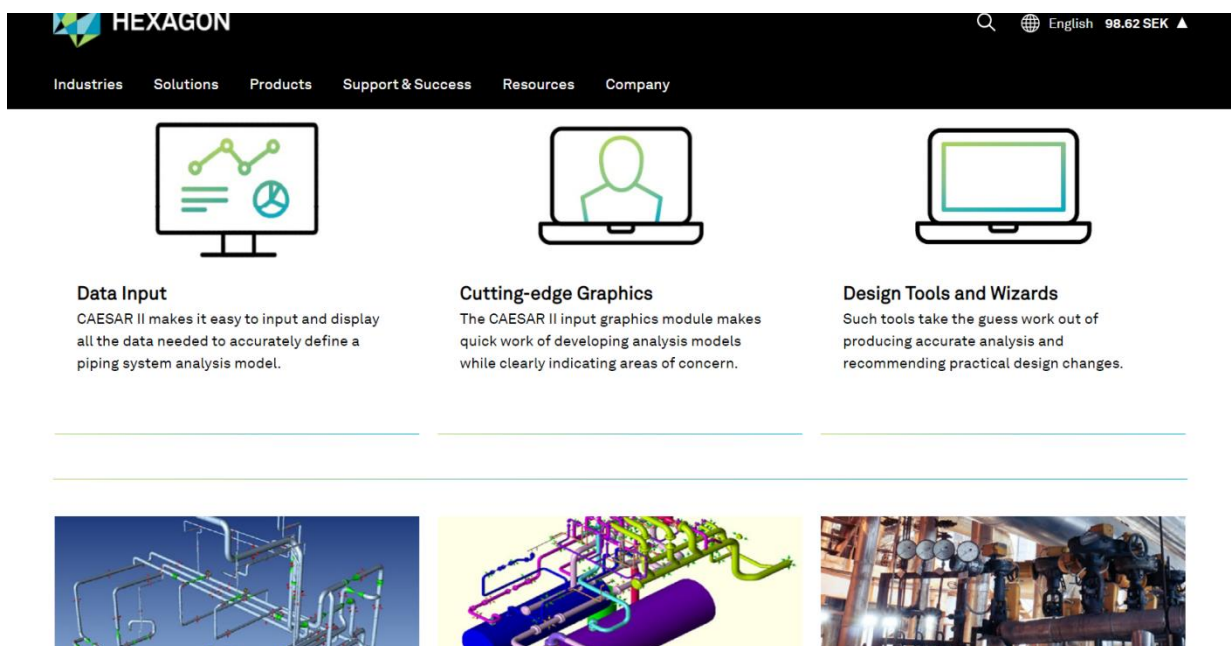


Рисунок 1.3 – Програмне забезпечення Caesar II [10], [13]

Програма Caesar II дозволяє імпортувати для комп'ютерною підтримки проєктування (Computer-Aided Design, CAD) для подальшого аналізу, незалежно від їх розміру [10], [13]. Caesar II виконує розрахунки статичних напружень, надаючи як графічні, так і числові рекомендації. Також вона має можливості для динамічного аналізу, включаючи розрахунок власних частот та аналіз ударних навантажень [10], [13].

Інтерактивний інтерфейс Caesar II дозволяє користувачам зручно переглядати моделі, пересуватися навколо них та анімувати динамічні реакції. Такі можливості роблять це програмне забезпечення важливим інструментом для інженерів, які займаються проєктуванням та розробкою у машинобудуванні [10], [13].

Серед особливостей програмного забезпечення Caesar II для підтримки процесу механічної обробки деталей можна виділити такі [10], [13]:

- можливість імпорту CAD-файлів: програма Caesar II дозволяє імпортувати файли у форматі CAD для здійснення аналізу та дослідження незалежно від розміру проєкту [10], [13];

- засоби аналізу напружень: програмне забезпечення Caesar II виконує розрахунки статичних напружень в трубопроводах і надає користувачам графічні та числові рекомендації [10], [13];

- засоби динамічного аналізу: програмне забезпечення Caesar II має великі можливості для динамічного аналізу, включаючи розрахунок власних частот і аналіз спектру ударів [10], [13];

- інтерактивний інтерфейс: інтерфейс Caesar II дозволяє користувачам зручно переглядати та взаємодіяти з моделями трубопроводів, анімувати динамічні реакції та здійснювати пересування навколо них [10], [13];

- засоби візуалізації результатів: програма надає можливість подати результати аналізу у зручному для сприйняття вигляді, що сприяє зрозумінню складних даних та допомагає в процесі прийняття рішень [10], [13];

- можливість застосування в різних галузях: програмне забезпечення Caesar II може використовуватися в автомобільній, аерокосмічній та інших

галузях промисловості, що підкреслює універсальність і широку застосовність цієї програми [10], [13];

– можливість виконання аналізу будь-якого розміру: програма Caesar II придатна для аналізу трубопроводів різних розмірів та складності, що дозволяє застосовувати її у різних проектах [10], [13].

Загалом, програма Caesar II володіє потужними функціями для аналізу та проектування у машинобудуванні, надаючи інженерам засоби для ретельної оцінки та вдосконалення проектів [10], [13].

Недоліками програмного забезпечення Caesar II можна вважати складність моделювання та аналізу, що може призводити до великих витрат часу на підготовку, налаштування та виконання моделювання. Програмне забезпечення Caesar II може вимагати відносно потужного обладнання для оптимальної продуктивності, що може бути проблемою для користувачів зі старими або менш потужними комп'ютерами [10], [13].

Порівняльну характеристику програмного забезпечення для підтримки процесу механічної обробки деталей наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для підтримки процесу механічної обробки деталей

Критерій порівняння	Fusion 360 [11]	Abaqus [12]	Caesar II [13]
Засоби інтегрованого середовища розробки	—	+	—
Безкоштовність	+—	—	—
Засоби фізичного аналізу	—	+	+—
Висока якість візуалізації	+	+—	+—
Підтримка різних платформ	+—	—	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для підтримки процесу

механічної обробки деталей. Проте деякі програмні засоби можуть не забезпечити достатньої точності у розрахунках, особливо якщо моделі, на яких вони базуються, мають деякі недоліки або невідомі параметри. Для точних розрахунків потрібні якісні дані про матеріали, інструменти, параметри обробки тощо, а деякі програмні засоби використовують застарілі або спрощені моделі, які не враховують частину з цих параметрів. Інші програмні засоби можуть бути досить складними у використанні, і вони вимагають від користувача високого рівня експертизи в області механічної обробки. Крім того, деякі програмні засоби для розрахунку параметрів механічної обробки деталей можуть бути платними або вимагати додаткових витрат на навчання персоналу. Тому актуальною є розробка програмного забезпечення для розрахунку параметрів механічної обробки деталей.

При розробці програмного забезпечення оптимізації для розрахунку параметрів механічної обробки деталей необхідно забезпечити такі функціональні вимоги:

- можливість введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей;
- можливість визначення припусків на механічну обробку деталей;
- подання результатів розрахунку припусків у вигляді карти припусків;
- можливість розрахунку виконавчих операційних розмірів при обробці торців;
- можливість розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки;
- можливість розрахунку економічного обґрунтування варіанта операції механічної обробки деталі.

1.3 Висновки за розділом 1

Визначено, що у наш час при виготовленні деталей машин та обладнання широко використовуються програмні засоби, оскільки програмні

засоби дозволяють проводити розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективну використання матеріалів та інструментів. Проаналізовано процес механічної обробки, що є важливою складовою виробництва багатьох виробів, визначено, що більшість сучасних процесів механічної обробки виконуються з використанням програмних засобів, які підтримують керування рухами і діями фрезерних верстатів. Проаналізовано процеси та операції виробництва та механічної обробки з використанням спеціалізованого програмного забезпечення. Проаналізовано різні види виробничих процесів, а також етапи оброблення деталей з використанням спеціалізованого програмного забезпечення. Відзначено, що при механічній обробці деталей машин та обладнання початкове креслення деталі відрізняється від готового креслення деталі тим, що на всіх оброблених поверхнях передбачені додаткові припуски для подальшої обробки. Точний розрахунок та вибір припусків та робочих припусків під час механічної обробки деталей є ключовим для розробки технологічних процесів, від яких залежать вартість, якість, тривалість експлуатації та надійність, що обґрунтовує доцільність розробки програмних засобів для розрахунку параметрів механічної обробки деталей, які дозволять автоматизувати відповідні процеси та здійснювати розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективне використання матеріалів та інструментів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу механічної обробки деталей. Проте деякі програмні засоби можуть не забезпечити достатньої точності у розрахунках, особливо якщо моделі, на яких вони базуються, мають деякі недоліки або невідомі параметри. Для точних розрахунків потрібні якісні дані про матеріали, інструменти, параметри обробки тощо, а деякі програмні засоби використовують застарілі або спрощені моделі, які не враховують частину з цих параметрів. Інші програмні засоби можуть бути досить складними у використанні, і вони вимагають від

користувача високого рівня експертизи в області механічної обробки. Крім того, деякі програмні засоби для розрахунку параметрів механічної обробки деталей можуть бути платними або вимагати додаткових витрат на навчання персоналу. Тому актуальною є розробка програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Сформульовано функціональні вимоги до програмного забезпечення для розрахунку параметрів механічної обробки деталей.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення оптимізації для розрахунку параметрів механічної обробки деталей було використано мову програмування C++ [14], [15].

C++ – це мова програмування, яка поєднує в собі зрозумілість та потужність об'єктно-орієнтованого підходу, і може бути порівняна з іншими мовами, такими як Perl, Ruby та Java [14], [15].

C++ є універсальною мовою програмування, відома своєю ефективністю та економічністю [14], [15]. Ці перелічені переваги C++ дозволяють створювати високоякісні програмні продукти майже будь-якого виду. Застосовуючи C++ як засіб програмування, можна отримувати швидкі й компактні програми. В багатьох випадках програму, написану на C++, можна порівняти за швидкістю з програмою, створеною за допомогою мови асемблера. Водночас програми на мові C++ мають кращу читабельність і їх легше відслідковувати [14], [15].

Хоча в C++ відсутні вбудовані мовні компоненти для розподілу пам'яті, управління екраном або керування процесом, середовище виконання C++ має набір бібліотек модулів проекту [14], [15], який реалізує аналогічні функції. Цей підхід дозволяє відокремити функції мови від специфічних особливостей процесора, на якому виконується кінцева програма [14], [15]. Жорстке визначення мови робить її незалежною від будь-яких деталей операційної системи чи апаратної платформи [14], [15]. Водночас розробники на мові C++ можуть додавати певні системні функції до бібліотеки, щоб скористатися певними можливостями апаратної платформи [14], [15].

Мова програмування C++ забезпечує повний набір операторів структурного програмування [14], [15].

Багато операцій в мові програмування C++ мають прямий відповідник в машинних командах, що дозволяє здійснювати безпосередній переклад у машинний код. Завдяки різним операціям можна вибирати різні набори, що сприяє зменшенню обсягу кінцевого коду [14], [15].

C++ містить препроцесор, який обробляє текстові файли перед компіляцією [14], [15]. Корисні можливості цієї мови програмування під час написання програм на C++ включають визначення програмних констант, заміну викликів функцій аналогічними, але більш швидкими макросами та умовну компіляцію. Препроцесор C++ можна використовувати для будь-якого текстового файлу [14], [15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	Ruby
Зручність для розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей	+	—	—
Безпека даних	+	+	+—
Робота з пам'яттю	+	—	—
Велика спільнота розробників	+	+	+—
Розширюваність	+	+	+—

Отже, для реалізації програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано мову програмування C++, яка дозволяє писати високоефективний код, що працює швидко навіть на обмежених ресурсах, надає різноманітні інструменти для програмування, такі як шаблони, перевантаження операторів, функціональні об'єкти та інші конструкції, що підвищують гнучкість програм та можливості розробки програмного забезпечення.

2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей

В якості середовища розробки для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей було обрано C++Builder [16], [17].

C++Builder є потужним інтегрованим середовищем розробки для програмування на мовах C і C++, яке використовується для створення як простих, так і професійних програм у операційних системах Windows, macOS, iOS та Android [16], [17].

Це середовище надає можливості навчання через приклади, навчальні, довідкові файли, а також підтримку розробників. C++ Builder є складовою частиною професійного середовища розробки, воно пропонує постійні випуски, нові функції, оновлення та підтримку, містить в собі бібліотеку для створення різних застосунків для Windows, а також бібліотеку для кросплатформеного інтерфейсу користувача. Крім того, існує безкоштовна версія для студентів, початківців і стартапів під назвою C++ Builder Community Edition [16], [17].

Інтегроване середовище розробки C++ Builder містить такі основні складові частини [16], [17]:

– вікно редактора коду та дизайнера форм: ця центральна панель середовища розробки C++ Builder містить редактор коду, який підтримує текстові можливості і має можливості підсвічування синтаксису, скасування дій та контекстну допомогу. Дизайнер форм генерує базовий код для інтерфейсу користувача, і зміни властивостей об'єктів автоматично відображаються у вихідних файлах [16], [17];

– вікно проєкту: тут відображаються та організовуються поточні групи проєктів та всі інші проєкти. Розробник може керувати проєктами, додавати та видаляти файли, компілювати їх [16], [17];

– вікно палітри є місцем, де містяться компоненти інтерфейсу користувача, які можна використовувати для розробки дизайну програми [16], [17];

– вікно структури є діаграмою, що відображає ієрархію елементів у відкритому вікні, такому як редактор коду або дизайнер форми [16], [17];

– вікно інспектора об'єктів: цей інструмент використовується для перевірки і зміни властивостей та подій обраних об'єктів [16], [17].

C++ Builder дозволяє створювати програми для різних операційних систем, таких як Windows, macOS, iOS і Android. Це полегшує розробку кросплатформених застосунків [16], [17].

C++ Builder є потужним засобом розробки, що надає розробникам розширення та бібліотеки для створення програм різного призначення, використовуючи як базис мову C++ [16], [17].

Інструмент C++ Builder надає приклади, навчальні посібники, довідкові файли та підтримку коду, що сприяє швидкому і ефективному навчанню для новачків [16], [17]. C++ Builder має багатий вибір компонентів і бібліотек, що значно прискорює розробку програм, включаючи інтерфейс користувача, мережеві функції, бази даних тощо [16], [17].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення ПЗ для розрахунку параметрів механічної обробки деталей

Критерій порівняння середовищ розробки	Середовище розробки		
	C++Builder	Visual Studio Code	Eclipse
Компоненти та бібліотеки	+	+—	+
Мови програмування	C++	Різні мови	Різні мови
Зручність редактору коду	+	+—	+
Безкоштовність	+—	+	+
Зручні інструменти для розробки на C++	+	+	—

Таким чином, для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано середовище розробки C++Builder, яке є потужним засобом розробки, що надає розробникам розширення та бібліотеки для створення програм різного призначення, використовуючи як базис мову C++.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей

З метою можливості розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей створимо відповідну модель.

При розробленні моделі взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей враховуємо те, що користувачі програми мають можливості введення параметрів заготовки для подальшого розрахунку параметрів механічної

обробки деталей, визначення припусків на механічну обробку деталей, подання результатів розрахунку припусків у вигляді карти припусків, розрахунку виконавчих операційних розмірів при обробці торців, розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки, розрахунку економічного обґрунтування варіанта операції механічної обробки деталі.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для розрахунку параметрів механічної обробки деталей можна описати таким чином:

- користувач програмного забезпечення для розрахунку параметрів механічної обробки деталей запускає програму;
- після запуску програми для розрахунку параметрів механічної обробки деталей користувачу відображається головна форма;
- користувач програми для розрахунку параметрів механічної обробки деталей бачить головну форму та вводить параметри заготовки для подальшого розрахунку параметрів механічної обробки деталей;
- користувач визначає припуски на механічну обробку деталей;
- користувач здійснює розрахунок економічного обґрунтування варіанта операції механічної обробки деталі;
- користувач завершує роботу з програмним забезпеченням для розрахунку параметрів механічної обробки деталей.

Враховуючи сценарії роботи користувачів при роботі з програмним забезпеченням для розрахунку параметрів механічної обробки деталей, модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей подамо у вигляді діаграми, зображеної на рис. 2.1.

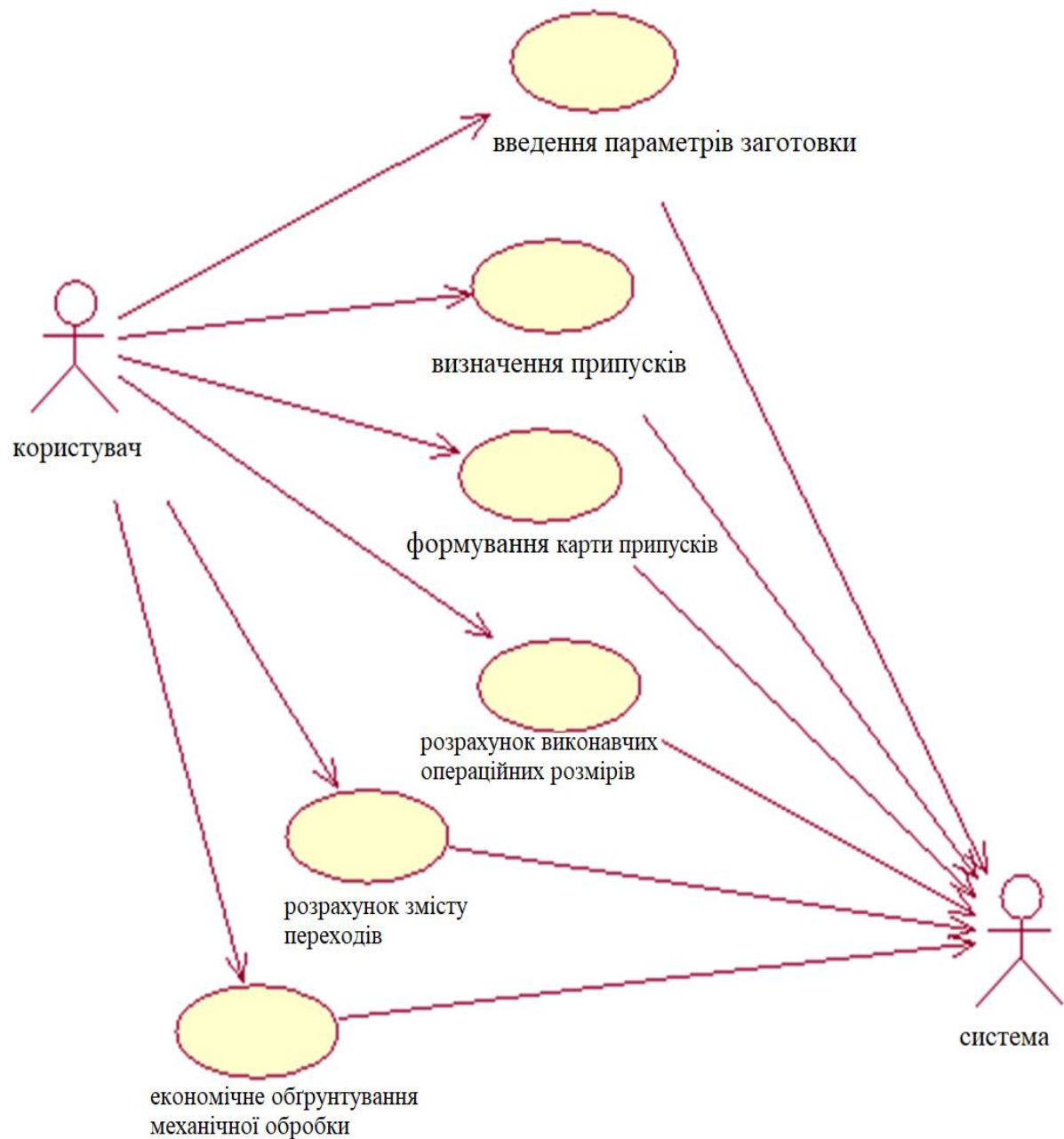


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей, яка дозволяє розроблювати програмне забезпечення для розрахунку параметрів механічної обробки деталей.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано мову програмування C++, яка дозволяє писати високоефективний код, що працює швидко навіть на обмежених ресурсах, надає різноманітні інструменти для програмування, такі як шаблони, перевантаження операторів, функціональні об'єкти та інші конструкції, що підвищують гнучкість програм та можливості розробки програмного забезпечення.

Для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано середовище розробки C++Builder, яке є потужним засобом розробки, що надає розробникам розширення та бібліотеки для створення програм різного призначення, використовуючи як базис мову C++.

Створено модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей, яка дозволяє розроблювати програмне забезпечення для розрахунку параметрів механічної обробки деталей.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для розрахунку параметрів механічної обробки деталей

Структуру програмного забезпечення для розрахунку параметрів механічної обробки деталей наведено на рис. 3.1.

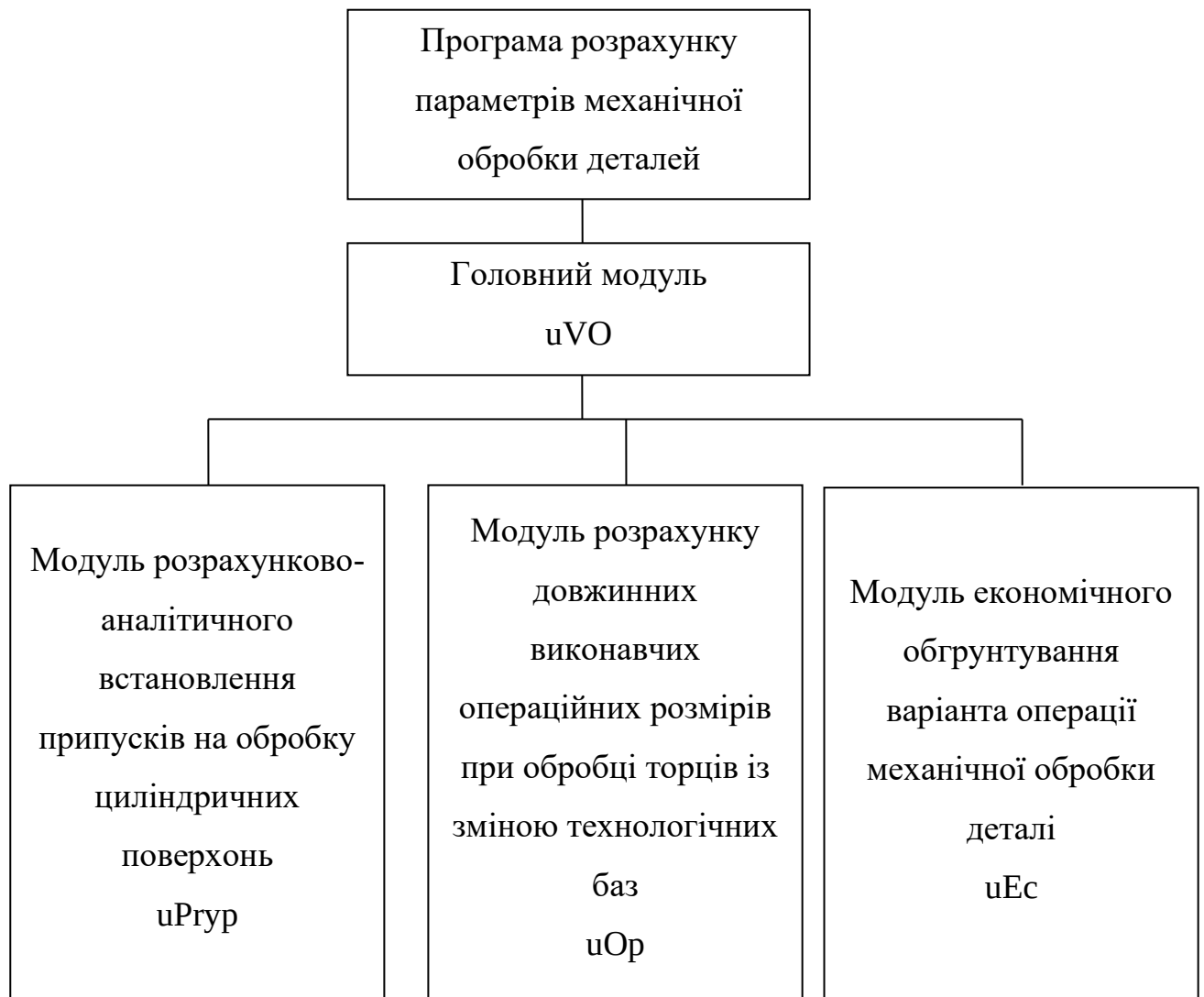


Рисунок 3.1 – Структура програмного забезпечення для розрахунку параметрів механічної обробки деталей

Як видно з рисунку, програмне забезпечення для розрахунку параметрів механічної обробки деталей складається з таких компонентів:

- головний модуль;
- модуль розрахунково-аналітичного встановлення припусків на обробку циліндричних поверхонь;
- модуль розрахунку довжинних виконавчих операційних розмірів при обробці торців із зміною технологічних баз;
- модуль економічного обґрунтування варіанта операції механічної обробки деталі.

Проаналізуємо призначення основних компонентів програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Головний модуль uVO є центральним модулем, через який виконується доступ до всіх методів, реалізованих в курсовому проекті. В ньому виконується обробка подій, які виникають під час роботи користувача з системою. Реагуючи на натиснення кнопок, система передає далі керування в модулі, що відповідають викликаному методу. Окрім того в даному модулі реалізується завантаження даних з файлу. Повернені з кожного окремого модуля результати виводяться на екран в даному модулі.

В модулі uPгур реалізовано метод розрахунково-аналітичного встановлення припусків на обробку циліндричних поверхонь, що базується на аналізі чинників, що впливають на припуск, який визначається з урахуванням складних його елементів. Передбачено розрахунок припусків по всім послідовно виконуваним технологічним переходам обробки даної поверхні з їх підсумовуванням для визначення загального припуску операційних виконавчих розмірів.

В модулі uOp використано теоретичне положення, що складається в тому, що всі розміри головної деталі, проміжні виконавчі операційні розміри і розміри вихідної заготовки зв'язані між собою сукупністю послідовно сформованих при обробці технологічних розмірних ланцюгів. Виявлення та побудування цих розмірних ланцюгів реалізовано в даному модулі.

При проектуванні технологічної операції має місце багатоваріантність можливих рішень. З усіх варіантів, що забезпечують якість і задану програму випуску деталей, необхідно вибрати варіант, який дає найбільший економічний ефект.

3.2 Функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей

Схема функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей наведена на рис. 3.2.

Робота з програмним забезпеченням для розрахунку параметрів механічної обробки деталей розпочинається з визначення режиму роботи та набору вхідних даних для відповідного режиму роботи.

Режим роботи може бути обрано один з наступних:

- установлення припусків на обробку поверхонь, де реалізовано функції для виконання методу встановлення припусків на обробку циліндричних поверхонь;
- розрахунок виконавчих операційних розмірів, що виконується при обробці торців із зміною технологічних баз;
- економічне обґрунтування варіанта операції механічної обробки деталі.

Кожний режим роботи системи реалізовано у вигляді окремого компоненту для забезпечення функціональності даного режиму. Після вибору режиму користувачем відбуваються відповідні розрахунки, результати яких виводяться на екран. Після цього користувач може або продовжити роботу з програмою, або завершити її виконання.

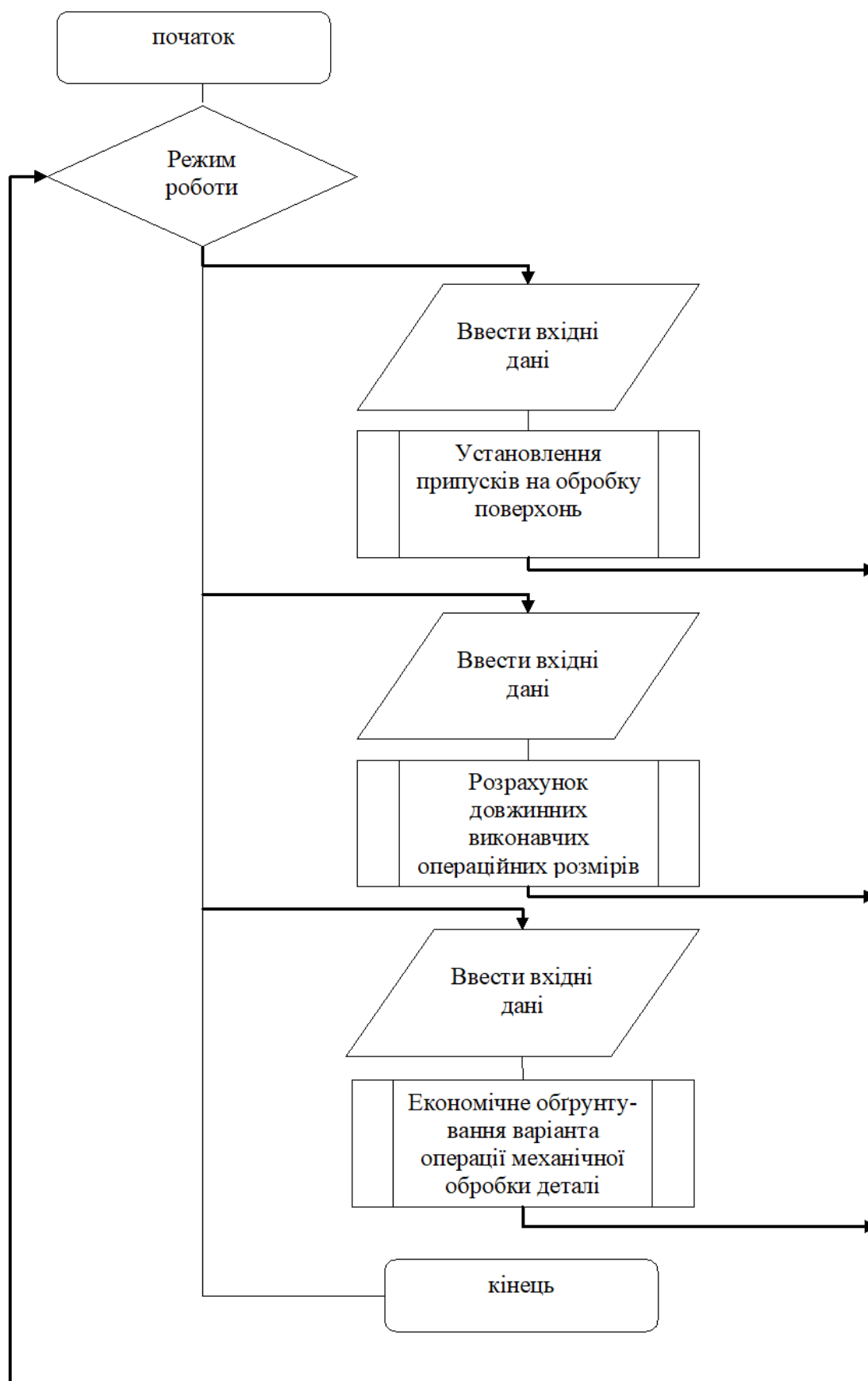


Рисунок 3.2 – Схема функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей

3.3 Проектування інтерфейсу програмного забезпечення для розрахунку параметрів механічної обробки деталей

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей виконано з урахуванням вимог технічного завдання.

Приклади розроблених інтерфейсних форм наведено на рис. 3.3-3.8.

Форму для введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей наведено на рис. 3.3.

The screenshot shows a software window titled 'VO 1.0' with a standard Windows-style title bar (blue with a close button). Below the title bar is a tab labeled 'Вхідні дані'. The main content area is divided into two sections. The first section, titled 'Заготовка' (Blank), contains four input fields: 'Маса' (Mass), 'Діаметр' (Diameter), 'Допуски' (Tolerances), and 'Тип поверхні' (Surface type). The second section, titled 'План обробки поверхні' (Surface processing plan), contains two dropdown menus: 'Тип обробки' (Processing type) and 'Спосіб обробки' (Processing method). At the bottom of the form are two buttons: 'Додати' (Add) and 'Розрахувати' (Calculate).

Рисунок 3.3 – Форма для введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей

Форму з картою розрахунку припусків відображено на рис. 3.4.

VO 1.0

Карта розрахунку припусків

План обробки	Rzi	hi	delEi	Ei	Z2min,мм	Допуск T ді,мм	Zін,мм	Zіmax,мм	dі,мм	Вик. опер. розмір
Заготовка	160	200	500	0	0	2000	0	0	38.43	38.4 +1,3 -0,7
Обтачування										
Чорнове	50	50	30	0	1720	390	2420	2810	36.01	36,01 -0,39
Обтачування										
Чистове	25	25	0	0	260	160	650	810	35.36	35,36 -0,16
Шліфування										
Чорнове	10	20	1	0	100	39	260	299	35.10	35,1 -0,039
Шліфування										
Чистове	5	15	0	0	62	30	101	131	35	35+0.05+0.02

Далі

Рисунок 3.4 – Форма з картою розрахунку припусків

Форму для розрахунку виконавчих операційних розмірів при обробці торців наведено на рис. 3.5.

VO 1.0

Вхідні дані

Розмір деталі

D1

D2

D3

A

B

Заготовка

Тип

Маса більш

до

Клас точності

План обробки торців

☐ Чорнова підрізка торця M

☐ Чистова підрізка торця M

☐ Чорнова підрізка торця P

☐ Чистова підрізка торця P

☐ Чорнова підрізка торця N

☐ Чистова підрізка торця N

☐ Попереднє шліфування торця M

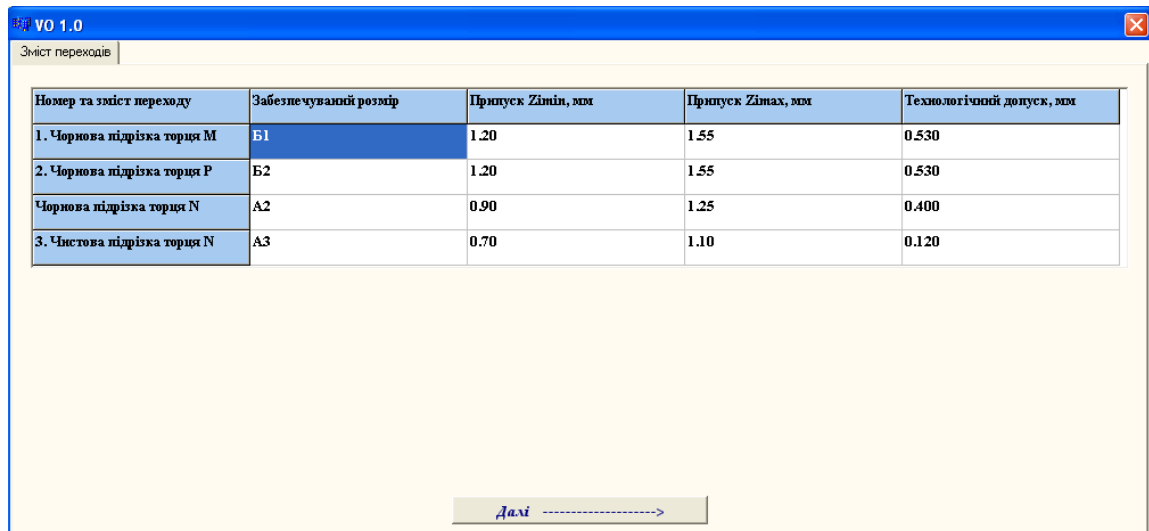
☐ Однократне шліфування торця P

☐ Чистове шліфування торця N

Підтвердити

Рисунок 3.5 – Форма для розрахунку виконавчих операційних розмірів при обробці торців

Інтерфейс форми розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки відображено на рис. 3.6.



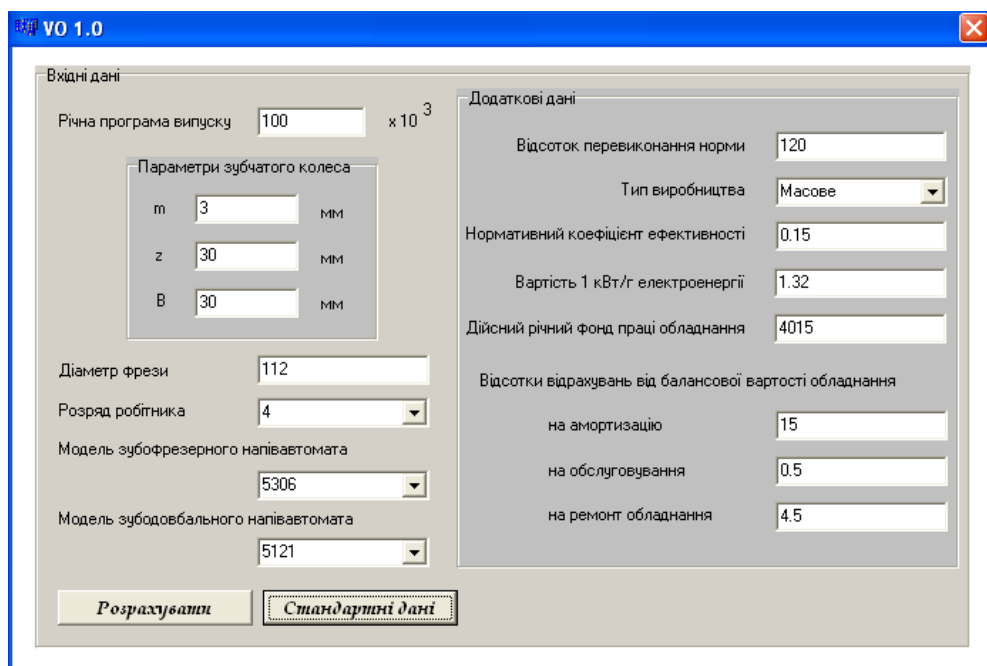
The screenshot shows a software window titled 'VO 1.0' with a tab labeled 'Зміст переходів'. It contains a table with 5 columns: 'Номер та зміст переходу', 'Забезпечуваний розмір', 'Припуск Zіmin, мм', 'Припуск Zіmax, мм', and 'Технологічний допуск, мм'. The table lists four transitions for rough turning of a part.

Номер та зміст переходу	Забезпечуваний розмір	Припуск Zіmin, мм	Припуск Zіmax, мм	Технологічний допуск, мм
1. Чорнова підрізка торця М	B1	1.20	1.55	0.530
2. Чорнова підрізка торця Р	B2	1.20	1.55	0.530
Чорнова підрізка торця N	A2	0.90	1.25	0.400
3. Чистова підрізка торця N	A3	0.70	1.10	0.120

Below the table is a button labeled 'Далі' with a right-pointing arrow.

Рисунок 3.6 – Інтерфейс форми розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки

Форму для введення даних для розрахунку економічного обґрунтування варіанта операції механічної обробки деталі наведено на рис. 3.7.



The screenshot shows a software window titled 'VO 1.0' with two main sections: 'Вхідні дані' (Input data) and 'Додаткові дані' (Additional data). The 'Вхідні дані' section includes fields for 'Річна програма випуску' (100 x 10³), 'Параметри зубчатого колеса' (m=3, z=30, B=30), 'Діаметр фрези' (112), 'Розряд робітника' (4), and two dropdown menus for machine models (5306 and 5121). The 'Додаткові дані' section includes fields for 'Відсоток перевиконання норми' (120), 'Тип виробництва' (Масове), 'Нормативний коефіцієнт ефективності' (0.15), 'Вартість 1 кВт/г електроенергії' (1.32), 'Дійсний річний фонд праці обладнання' (4015), and three fields for percentages of depreciation (15), maintenance (0.5), and repair (4.5). At the bottom are buttons for 'Розрахувати' and 'Стандартні дані'.

Рисунок 3.7 – Форма для введення даних для розрахунку економічного обґрунтування варіанта операції механічної обробки деталі

Форму з виведенням результатів розрахунку економічного обґрунтування варіанта операції механічної обробки деталі наведено на рис. 3.8.

Статті/Режим	Зубофрезерування	Зубодовбання
Основний час на операцію	6.28	3.35
Норма штучного часу	7.02	3.79
Кількість верстатів	3	2
Верстатоемкість	5.85	3.16
Витрати на		
- заробітну плату	1.26	1.26
- електроенергію	0.08	0.04
- амортизацію обладнання	0.38	0.43
- обслуговування обладнання	0.01	0.01
- ремонт обладнання	0.11	0.13
- інструмент	0.245	0.590
Технологічна собівартість операції	0.1220	0.0778
Економічна доцільність	-	+

Економічний ефект

Змінити вхідні дані

Обрати іншу роботу

Рисунок 3.8 – Форма з виведенням результатів розрахунку економічного обґрунтування варіанта операції механічної обробки деталі

3.4 Висновки за розділом 3

Розроблено програмне забезпечення для розрахунку параметрів механічної обробки деталей. Запропоновано структуру та описано основні модулі програмного забезпечення для розрахунку параметрів механічної обробки деталей. Визначено, що основними компонентами програми для розрахунку параметрів механічної обробки деталей є: головний модуль,

модуль розрахунково-аналітичного встановлення припусків на обробку циліндричних поверхонь, модуль розрахунку виконавчих операційних розмірів при обробці торців із зміною технологічних баз, модуль економічного обґрунтування варіанта операції механічної обробки деталі.

Описано функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для розрахунку параметрів механічної обробки деталей. Програма написана на мові C++, яка дозволяє писати високоефективний код, що працює швидко навіть на обмежених ресурсах, надає різноманітні інструменти для програмування, такі як шаблони, перевантаження операторів, функціональні об'єкти та інші конструкції, що підвищують гнучкість програм та можливості розробки програмного забезпечення. В якості середовища розробки для створення програми обрано C++Builder, яке є потужним засобом розробки, що надає розробникам розширення та бібліотеки для створення програм різного призначення, використовуючи як базис мову C++.

Програмне забезпечення для розрахунку параметрів механічної обробки деталей забезпечує виконання таких функцій:

- можливість введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей;
- можливість визначення припусків на механічну обробку деталей;
- подання результатів розрахунку припусків у вигляді карти припусків;
- можливість розрахунку виконавчих операційних розмірів при обробці торців;
- можливість розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки;
- можливість розрахунку економічного обґрунтування варіанта операції механічної обробки деталі.

Для експлуатації програмного забезпечення для розрахунку параметрів механічної обробки деталей необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 500 MHz і більше;

- вільне місце на диску 100 Мб або більше;
- оперативна пам'ять об'ємом 128 Мб і більше;
- операційна система сімейства Windows.

Функціональні характеристики розробленого програмного забезпечення для розрахунку параметрів механічної обробки деталей наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для розрахунку параметрів механічної обробки деталей наведено у додатку Б.

4.2 Характеристики програми для розрахунку параметрів механічної обробки деталей

Програмне забезпечення для розрахунку параметрів механічної обробки деталей може працювати безперервно, в залежності від потреб користувача.

Програмне забезпечення для розрахунку параметрів механічної обробки деталей може використовуватись під керівництвом операційної системи сімейства Windows.

Для коректного функціонування програми для розрахунку параметрів механічної обробки деталей вміст директорії з програмним забезпеченням не можна змінювати.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для розрахунку параметрів механічної обробки деталей необхідно завантажити відповідні файли, а також відкрити основний файл pVO.exe стандартним способом використовуваної операційної системи.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для розрахунку параметрів механічної обробки деталей є параметри заготовки (маса, діаметр, тип поверхні та ін.), клас точності, план обробки торців, річна програма випуску та інші.

Вихідними даними програмного забезпечення для розрахунку параметрів механічної обробки деталей є карта розрахунку припусків, відповідні значення припусків, зміст переходів, номінальні розміри та виконавчі операційні розміри та інші.

4.3.3 Повідомлення

Користувач програмного забезпечення для розрахунку параметрів механічної обробки деталей може отримати різні повідомлення, зокрема, повідомлення про некоректно введені вхідні дані.

4.4 Виконання програмного забезпечення для розрахунку параметрів механічної обробки деталей

Після запуску програмного забезпечення для розрахунку параметрів механічної обробки деталей користувачу виводиться головна форма програми, де він може обрати один з можливих варіантів функціонування програми:

- розрахунок параметрів механічної обробки деталей;
- розрахунок виконавчих операційних розмірів при обробці торців;
- економічного обґрунтування варіанта операції механічної обробки деталі.

При виборі першого варіанта головного меню з'явиться форма для введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей (рис. 4.1). Користувачу необхідно ввести

інформацію щодо заготовки (маса, діаметр, допуски, тип поверхні), а також визначити тип та спосіб обробки.

The screenshot shows a software window titled 'VO 1.0' with a standard Windows-style title bar (blue with a close button). The window contains a tab labeled 'Вхідні дані' (Input data). Below the tab, there are two main sections with green headers: 'Заготовка' (Blank) and 'План обробки поверхні' (Surface processing plan). The 'Заготовка' section includes four input fields: 'Маса' (Mass), 'Діаметр' (Diameter), 'Допуски' (Tolerances), and 'Тип поверхні' (Surface type). The 'Допуски' field is split into two lines. The 'План обробки поверхні' section includes two dropdown menus: 'Тип обробки' (Processing type) and 'Спосіб обробки' (Processing method). At the bottom of the 'План обробки поверхні' section is a button labeled 'Додати' (Add). At the very bottom of the window is a large button labeled 'Розрахувати' (Calculate).

Рисунок 4.1 – Форма для введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей

При натисканні на кнопку "Додати", програмне забезпечення надає можливість додавання даних про черговий етап плану обробки поверхні (рис. 4.2). Таким чином можна ввести інформацію щодо всіх етапів планової обробки поверхні заготовки.

The screenshot shows a software window titled 'VO 1.0' with a standard Windows-style title bar (blue with a close button). The window has a tab labeled 'Вхідні дані' (Input data). Inside, there are two main sections with green backgrounds:

- Заготовка** (Workpiece):
 - Маса** (Mass): text input field containing '3.8'.
 - Діаметр** (Diameter): text input field containing '35'.
 - Допуски** (Tolerances): two stacked text input fields containing '+0.05' and '+0.02'.
 - Тип поверхні** (Surface type): dropdown menu with 'Вал' (Shaft) selected.
- План обробки поверхні** (Surface processing plan):
 - Тип обробки** (Processing type): dropdown menu with 'Шліфування' (Grinding) selected.
 - Спосіб обробки** (Processing method): dropdown menu with 'Чистове' (Finish) selected.
 - A button labeled **Додати** (Add) in green text.

At the bottom of the window is a large button labeled **Розрахувати** (Calculate) in green text.

Рисунок 4.2 – Введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей

Після натискання на кнопку "Розрахувати" виводяться результати розрахунку параметрів механічної обробки деталей у вигляді карти розрахунку припусків (рис. 4.3).

При натисканні на кнопку "Далі", програмне забезпечення запускає наступний етап розрахунків, на якому на екран буде виведено значення наступних припусків.

План обробки	Rzi	hi	delEi	Ei	Z2min,мм	ДопустkT,мм	Zin,мм	Zimax,мм	d,мм	Вик. опер. розмір
Заготовка	160	200	500	0	0	2000	0	0	38.43	38.4 +1,3 -0,7
Обтачування										
Чорнове	50	50	30	0	1720	390	2420	2810	36.01	36,01 -0,39
Обтачування										
Чистове	25	25	0	0	260	160	650	810	35.36	35,36 -0,16
Шліфування										
Чорнове	10	20	1	0	100	39	260	299	35.10	35,1 -0,039
Шліфування										
Чистове	5	15	0	0	62	30	101	131	35	35+0.05+0.02

Далі ----->

Рисунок 4.3 – Форма з картою розрахунку припусків

При виборі другого варіанту головного меню програми відображається форма для введення параметрів з метою подальшого розрахунку виконавчих операційних розмірів при обробці торців (рис. 4.4).

Вхідні дані

Розмір деталі

D1

D2

D3

A

B

Заготовка

Тип

Маса більш

до

Клас точності

План обробки торців

☐ Чорнова підрізка торця М

☐ Чистова підрізка торця М

☐ Чорнова підрізка торця Р

☐ Чистова підрізка торця Р

☐ Чорнова підрізка торця N

☐ Чистова підрізка торця N

☐ Попереднє шліфування торця М

☐ Однократне шліфування торця Р

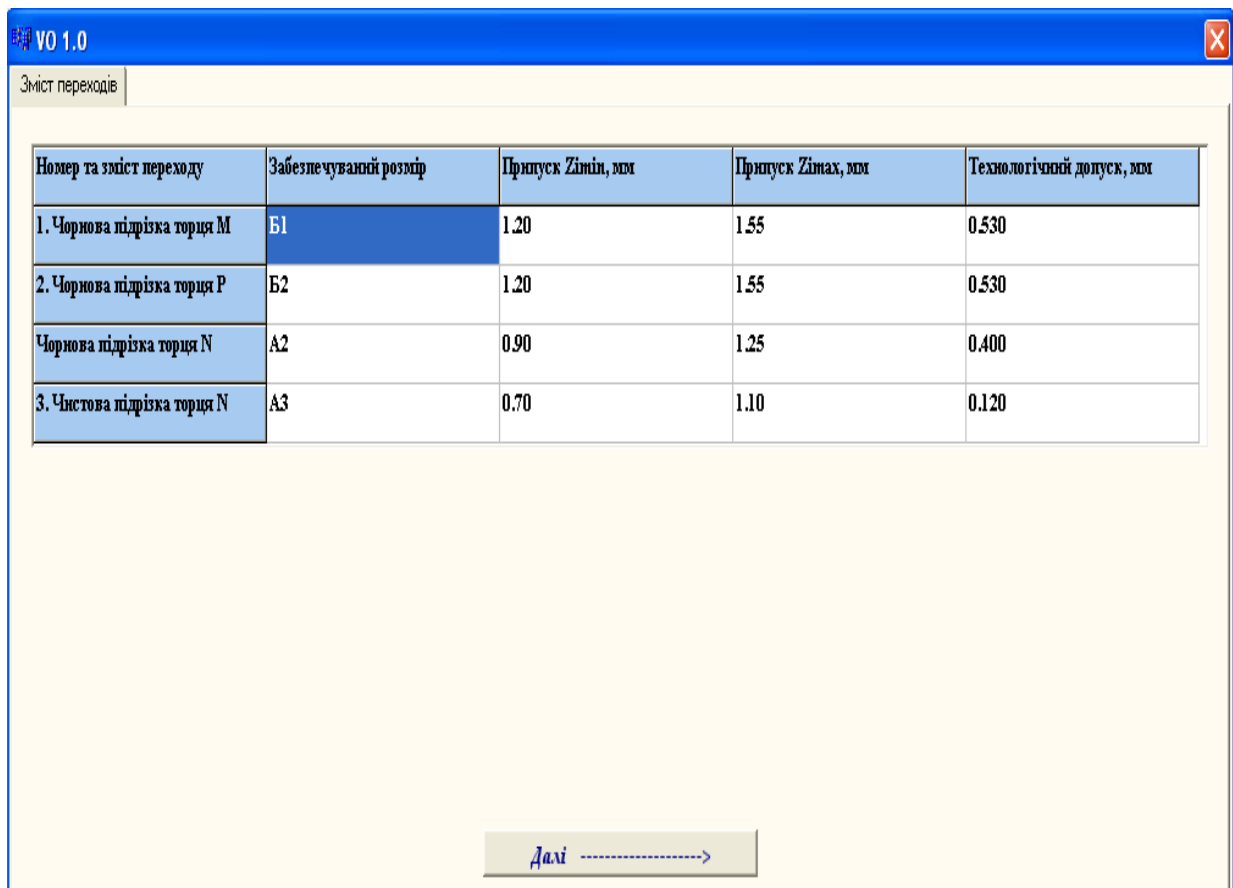
☐ Чистове шліфування торця N

Підтвердити

Рисунок 4.4 – Форма для розрахунку виконавчих операційних розмірів при обробці торців

Тут користувач вводить розміри дедалі згідно креслення, параметри заготовки (тип заготовки, інтервал маси заготовки та клас точності в залежності від типу заготовки), а також обирає план обробки торців.

Після натискання відповідної кнопки відбувається розрахунок змісту переходів у процесі механічної обробки. Форму подання у табличному вигляді змісту переходів у процесі механічної обробки відображено на рис. 4.5.



Номер та зміст переходу	Забезпечуваний розмір	Припуск Zimin, мм	Припуск Zimax, мм	Технологічний допуск, мм
1. Чорнова підрізка торця М	Б1	1.20	1.55	0.530
2. Чорнова підрізка торця Р	Б2	1.20	1.55	0.530
Чорнова підрізка торця N	A2	0.90	1.25	0.400
3. Чистова підрізка торця N	A3	0.70	1.10	0.120

Далі ----->

Рисунок 4.5 – Інтерфейс форми подання у табличному вигляді змісту переходів у процесі механічної обробки

При натисканні на кнопку "Далі" буде здійснено наступний етап розрахунку та виведено номінальні розміри для процесу механічної обробки (рис. 4.6).

Розмір	Величина, мм
A0	48.90
A1	47.35
A2	46.10
B0	93.10
B1	91.55

Далі ----->

Рисунок 4.6 – Виведення номінальних розмірів

Після натискання користувачем кнопки "Далі" буде здійснено черговий етап розрахунків, а результати буде подано у вигляді таблиці з інформацією про виконавчі операційні розміри при обробці торців (рис.4.7).

Операція	Розмір	Величина, мм	Технологічний допуск, мм
Чорнова підрізка торця М	B1	91.55	0.530
Чорнова підрізка торця Р	B2	90.000	0.530
Чорнова підрізка торця N	A2	46.10	0.400
Чистова підрізка торця N	A3	45.000	0.120

Введення даних

Вибрати іншу роботу

Рисунок 4.7 – Таблиця з інформацією про виконавчі операційні розміри при обробці торців

При виборі третього варіанту головного меню програми відображається форма для введення параметрів з метою подальшого економічного обґрунтування варіанта операції механічної обробки деталі (рис. 4.8).

The screenshot shows a software window titled 'VO 1.0' with a blue title bar and a close button. The window is divided into two main sections: 'Вхідні дані' (Input data) on the left and 'Додаткові дані' (Additional data) on the right. At the bottom, there are two buttons: 'Розрахувати' (Calculate) and 'Стандартні дані' (Standard data).

Вхідні дані (Input data):

- Річна програма випуску: 100 $\times 10^3$
- Параметри зубчатого колеса (Gear parameters):**
 - m: 3 мм
 - z: 30 мм
 - B: 30 мм
- Діаметр фрези: 112
- Розряд робітника: 4
- Модель зубофрезерного напівавтомата: 5306
- Модель зубодовбального напівавтомата: 5121

Додаткові дані (Additional data):

- Відсоток перевиконання норми: 120
- Тип виробництва: Масове
- Нормативний коефіцієнт ефективності: 0.15
- Вартість 1 кВт/г електроенергії: 1.32
- Дійсний річний фонд праці обладнання: 4015
- Відсотки відрахувань від балансової вартості обладнання (Depreciation percentages from the balance value of the equipment):**
 - на амортизацію: 15
 - на обслуговування: 0.5
 - на ремонт обладнання: 4.5

Рисунок 4.8 – Форма для введення даних для розрахунку економічного обґрунтування варіанта операції механічної обробки деталі

Тут користувачу необхідно ввести інформацію щодо річної програми випуску, параметрів виробу та інструменти, розряд робітника. Також вводяться додаткові дані, зокрема, відсоток перевиконання норми, тип виробництва, нормативний коефіцієнт ефективності, вартість електроенергії, дійсний річний фонд праці обладнання, а також відрахування на амортизацію, обслуговування та ремонт обладнання.

Після натискання на кнопку "Розрахувати" відображається форма з результатами розрахунку економічного обґрунтування варіанта операції механічної обробки деталі (рис. 4.9).

VO 1.0

Статті/Режим	Зубофрезерування	Зубодовбання
Основний час на операцію	6.28	3.35
Норма штучного часу	7.02	3.79
Кількість верстатів	3	2
Верстатоемкість	5.85	3.16
Витрати на		
- заробітну плату	1.26	1.26
- електроенергію	0.08	0.04
- амортизацію обладнання	0.38	0.43
- обслуговування обладнання	0.01	0.01
- ремонт обладнання	0.11	0.13
- інструмент	0.245	0.590
Технологічна собівартість операції	0.1220	0.0778
Економічна доцільність	-	+

Економічний ефект

Змінити вхідні дані

Обрати іншу роботу

Рисунок 4.9 – Форма з результатами розрахунку економічного обґрунтування варіанта операції механічної обробки деталі

При необхідності користувач програмного забезпечення може змінити вхідні дані, натиснувши відповідну кнопку.

4.5 Тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей

Виконано тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Для тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей були використані дані, які використовуються в

процесі виробництва. Ці дані включають такі характеристики, як маса, діаметр, допуски та тип поверхні деталі, а також плани обробки поверхні. Для аналізу циліндричних поверхонь використовувався розрахунково-аналітичний метод визначення припусків. Також враховувалися розмір деталі, тип заготовки, інтервал маси заготовки, клас точності, плани обробки торців з урахуванням технологічних баз. Також враховувались додаткові фактори, такі як відсоток перевищення норми, тип виробництва, коефіцієнт ефективності, вартість електроенергії, фонд праці обладнання, відсотки відрахувань для амортизації, обслуговування та ремонту обладнання. Ці дані були використані для економічного обґрунтування оптимального варіанта механічної обробки деталі.

За результатами тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей встановлено, що програма дозволяє розв'язувати поставлені завдання та може використовуватися за призначенням.

4.6 Висновки за розділом 4

Описано програмне забезпечення для розрахунку параметрів механічної обробки деталей. Виконано тестування розробленого програмного забезпечення для розрахунку параметрів механічної обробки деталей. Результати тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей показали, що розроблена програма дозволяє розв'язувати поставлені завдання та може використовуватися за призначенням.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для механічного оброблення деталей.

Визначено, що у наш час при виготовленні деталей машин та обладнання широко використовуються програмні засоби, оскільки програмні засоби дозволяють проводити розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективну використання матеріалів та інструментів. Проаналізовано процес механічної обробки, що є важливою складовою виробництва багатьох виробів, визначено, що більшість сучасних процесів механічної обробки виконуються з використанням програмних засобів, які підтримують керування рухами і діями фрезерних верстатів. Проаналізовано процеси та операції виробництва та механічної обробки з використанням спеціалізованого програмного забезпечення. Проаналізовано різні види виробничих процесів, а також етапи оброблення деталей з використанням спеціалізованого програмного забезпечення. Відзначено, що при механічній обробці деталей машин та обладнання початкове креслення деталі відрізняється від готового креслення деталі тим, що на всіх оброблених поверхнях передбачені додаткові припуски для подальшої обробки. Точний розрахунок та вибір припусків та робочих припусків під час механічної обробки деталей є ключовим для розробки технологічних процесів, від яких залежать вартість, якість, тривалість експлуатації та надійність, що обґрунтовує доцільність розробки програмних засобів для розрахунку параметрів механічної обробки деталей, які дозволять автоматизувати відповідні процеси та здійснювати розрахунки параметрів механічної обробки з високою точністю, забезпечуючи ефективне використання матеріалів та інструментів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу механічної

обробки деталей. Проте деякі програмні засоби можуть не забезпечити достатньої точності у розрахунках, особливо якщо моделі, на яких вони базуються, мають деякі недоліки або невідомі параметри. Для точних розрахунків потрібні якісні дані про матеріали, інструменти, параметри обробки тощо, а деякі програмні засоби використовують застарілі або спрощені моделі, які не враховують частину з цих параметрів. Інші програмні засоби можуть бути досить складними у використанні, і вони вимагають від користувача високого рівня експертизи в області механічної обробки. Крім того, деякі програмні засоби для розрахунку параметрів механічної обробки деталей можуть бути платними або вимагати додаткових витрат на навчання персоналу. Тому актуальною є розробка програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Сформульовано функціональні вимоги до програмного забезпечення для розрахунку параметрів механічної обробки деталей.

Для реалізації програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано мову програмування C++, яка дозволяє писати високоефективний код, що працює швидко навіть на обмежених ресурсах, надає різноманітні інструменти для програмування, такі як шаблони, перевантаження операторів, функціональні об'єкти та інші конструкції, що підвищують гнучкість програм та можливості розробки програмного забезпечення.

Для створення програмного забезпечення для розрахунку параметрів механічної обробки деталей обрано середовище розробки C++Builder, яке є потужним засобом розробки, що надає розробникам розширення та бібліотеки для створення програм різного призначення, використовуючи як базис мову C++.

Новизна роботи полягає в тому, що запропоновано модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей, та дозволяє розроблювати програмне забезпечення для розрахунку параметрів механічної обробки деталей.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для розрахунку параметрів механічної обробки деталей.

Запропоновано структуру та описано основні модулі програмного забезпечення для розрахунку параметрів механічної обробки деталей. Визначено, що основними компонентами програми для розрахунку параметрів механічної обробки деталей є: головний модуль, модуль розрахунково-аналітичного встановлення припусків на обробку циліндричних поверхонь, модуль розрахунку виконавчих операційних розмірів при обробці торців із зміною технологічних баз, модуль економічного обґрунтування варіанта операції механічної обробки деталі.

Описано функціонування програмного забезпечення для розрахунку параметрів механічної обробки деталей. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей.

Виконано тестування розробленого програмного забезпечення для розрахунку параметрів механічної обробки деталей. Результати тестування програмного забезпечення для розрахунку параметрів механічної обробки деталей показали, що розроблена програма дозволяє розв'язувати поставлені завдання та може використовуватися за призначенням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. <https://testbook.com/mechanical-engineering/types-of-manufacturing-processes> [Electronic resource]. – Access mode : <https://www.rapiddirect.com/blog/types-of-machining-operations/>.
2. Types of Manufacturing Processes [Electronic resource]. – Access mode : <https://www.mechanicaleducation.com/types-of-manufacturing-processes>.
3. Metal Parts Processing [Electronic resource]. – Access mode : https://metalpartss.com/Metal_Parts_Processing/.
4. The production process of mechanical parts [Electronic resource]. – Access mode : <https://www.yontongroup.com/the-production-process-of-mechanical-parts/>.
5. Types of Manufacturing Processes [Electronic resource]. – Access mode : <https://testbook.com/mechanical-engineering/types-of-manufacturing-processes>.
6. Mechanical computing [Electronic resource]. – Access mode : <https://www.nature.com/articles/s41586-021-03623-y>.
7. Top 8 Mechanical Engineering Software [Electronic resource]. – Access mode : <https://fixthephoto.com/best-mechanical-engineering-software.html>.
8. The 10 Best Mechanical Engineering Software in 2023 [Electronic resource]. – Access mode : <https://www.machinemfg.com/top-10-mechanical-engineering-design-software/>.
9. Mechanical Engineering Software [Electronic resource]. – Access mode : <https://www.g2.com/software/mechanical-engineering>.
10. The Best Mechanical Engineering Software for Every Application [Electronic resource]. – Access mode : <https://www.cncsourced.com/software/best-mechanical-engineering-software/>.
11. What is Fusion 360 [Electronic resource]. – Access mode : <https://www.autodesk.in/products/fusion-360/free-trial>.
12. Abaqus [Electronic resource]. – Access mode : <https://www.3ds.com/edu/education/students/solutions/abaqus-le>.

13. Caesar II [Electronic resource]. – Access mode :
<https://hexagon.com/products/caesar-ii>.

14. C++ Compiler [Electronic resource]. – Access mode :
<https://www.embarcadero.com/free-tools/ccompiler>.

15. Pattern Program In C++ [Electronic resource]. – Access mode :
<https://www.tutorialsonight.com/pattern-program-in-c%2B%2B>.

16. C++Builder [Electronic resource]. – Access mode :
<https://www.embarcadero.com/products/cbuilder/>.

17. C++ Builder or Visual Studio [Electronic resource]. – Access mode :
<https://stackoverflow.com/questions/4234434/c-builder-or-visual-studio>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для розрахунку параметрів механічної обробки деталей.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів розрахунку параметрів механічної обробки деталей», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

A.2 Призначення розробки

Програмний продукт призначений для забезпечення та підтримки процесу розрахунку параметрів механічної обробки деталей.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для розрахунку параметрів механічної обробки деталей забезпечує виконання таких функцій:

- можливість введення параметрів заготовки для подальшого розрахунку параметрів механічної обробки деталей;
- можливість визначення припусків на механічну обробку деталей;
- подання результатів розрахунку припусків у вигляді карти припусків;
- можливість розрахунку виконавчих операційних розмірів при обробці торців;
- можливість розрахунку та подання у табличному вигляді змісту переходів у процесі механічної обробки;

– можливість розрахунку економічного обґрунтування варіанта операції механічної обробки деталі.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для розрахунку параметрів механічної обробки деталей повинен бути зручним для користувачів.

А.3.3 Вимоги до надійності

Програмне забезпечення для розрахунку параметрів механічної обробки деталей повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для розрахунку параметрів механічної обробки деталей необхідна наявність персонального комп'ютера у користувача.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для розрахунку параметрів механічної обробки деталей необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 500 MHz і більше;
- вільне місце на диску 100 Мб або більше;
- оперативна пам'ять об'ємом 128 Мб і більше;
- операційна система сімейства Windows.

A.3.6 Вимоги до маркування і пакування

Програма для розрахунку параметрів механічної обробки деталей може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для розрахунку параметрів механічної обробки деталей».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для розрахунку параметрів механічної обробки деталей є параметри заготовки (маса, діаметр, тип поверхні та ін.), клас точності, план обробки торців, річна програма випуску та інші.

Вихідними даними програмного забезпечення для розрахунку параметрів механічної обробки деталей є карта розрахунку припусків, відповідні значення припусків, зміст переходів, номінальні розміри та виконавчі операційні розміри та інші.

ДОДАТОК Б
Фрагмент тексту програми

```

void TrmkVyb::prk6Click(TObject *Sender)
{
    hnfl4->Show();
    ftmvyn->Close();
}
void trmkVyb::prk5Click(TObject *Sender)
{
    hnfl5->Show();
    ftmvyn->Close();
}
void trmkVyb::prk6Click(TObject *Sender)
{
    hnfl6->Show();
    ftmvyn->Close();
}
void trmkVyb::btnCloseClick(TObject *Sender)
{
    drtzas->btnCancelClick(drtzas);
}
void trmkVyb::FormCreate(TObject *Sender)
{
    if ((Sender==hnfl4) || (Sender==hnfl5) || (Sender==hnfl6))
        {ftmvyn->Show();}
}

TrmkL4 *hnfl4;
int kol=0;
float Hntbf16[8][3]={ {25,0.2,1.25},
                      {50,0.15,0.75},
                      {80,0.12,0.75},
                      {360,0,0},
                      {500,0,0}};
float Hntbf29[6]={0.05,0.06,0.05,0.04,0.03,0.02};
float Hntbf12[6][3]={ {0.25,80,150},
                      {4,160,200},
                      {100,320,350},
                      {200,400,400}};
float Hntbf25[5][2]={ {50,50},
                      {25,25},
                      {2.5,5}};
float Hntbf21[6]={2000,390,160,39,30,0};
typedef float Hntbf23[18][28];
Hntbf23
float Radz[20],h[20],T[20],dlERT[20],ERT[20],Zmn[20],Zxn[20],Zmx[20],D[20];
float crt2,cs2;
trmkL4::TrmkL4(TComponent* Vldc)
    : TForm(Vldc)
{
    for (int i=0;i<20;i++)
        { ERT[i]=0;
          dlERT[i]=0;
        }
}
void trmkL4::FormClose(TObject *Sender, TCloseAction &Action)
{
    ftmvyn->FormCreate(hnfl4);
    kol=0;
    tbinpt->TabVisible=true;
    tbinpt->Visible=true;
    tbmp->TabVisible=false;
    tbmp->Visible=false;
    tbkr42->TabVisible=false;
    tbkr42->Visible=false;
}

```



```

        break;}
    case 1:{bcdmtp->Items->Add("Чернове");
        bcdmtp->Items->Add("Чистве");
        break;}
    case 2:{bcdmtp->Items->Add("Одинарне");
        break;}
    case 3:{bcdmtp->Items->Add("Одинарне");
        bcdmtp->Items->Add("Чернове");
        bcdmtp->Items->Add("Чистве");
        bcdmtp->Items->Add("Фінішне");
        break;}
    default:{bcdmtp->ItmIndex=-1;
        bcdmtp->Clear();
        }
    }
}
if (vbyr->ItmIndex==1)
{
    switch (bcmtp->ItmIndex) {
        case 0:{bcdmtp->Items->Add("Одинарне");
            bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("НапівЧистве");
            bcdmtp->Items->Add("Чистве");
            break;}
        case 1:{bcdmtp->Items->Add("Одинарне");
            bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("НапівЧистве");
            bcdmtp->Items->Add("Чистве");
            break;}
        case 2:{bcdmtp->Items->Add("Одинарне");
            bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("Чистве");
            bcdmtp->Items->Add("Фінішне");
            break;}
        default:{bcdmtp->ItmIndex=-1;
            bcdmtp->Clear();
            }
        }
    }
if (vbyr->ItmIndex==2)
{
    switch (bcmtp->ItmIndex) {
        case 0:{bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("Чистве");
            break;}
        case 1:{bcdmtp->Items->Add("Одинарне");
            break;}
        case 2:{bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("Чистве");
            break;}
        case 3:{bcdmtp->Items->Add("Одинарне");
            bcdmtp->Items->Add("Чернове");
            bcdmtp->Items->Add("Чистве");
            bcdmtp->Items->Add("Фінішне");
            break;}
        default:{bcdmtp->ItmIndex=-1;
            bcdmtp->Clear();
            }
        }
    }
if (vbyr->ItmIndex==-1)
{bcmtp->ItmIndex=-1;
    bcdmtp->ItmIndex=-1;
    bcmtp->Clear();
    bcdmtp->Clear();
}
}

```

```

}
void trmkL4::bbtAddPClick(TObject *Sender)
{
    int t=0;
    if (vbyr->ItmIndex==0)
        { if (bcmtpr->ItmIndex==0)
            {if (bcdmtp->ItmIndex==1) {t=1;}
            else {if (bcdmtp->ItmIndex==2) {t=2;}
                else {ShowMessage("Введено не всі інформацію");}
            }
        }
        else { if (bcmtpr->ItmIndex==3)
            {if (bcdmtp->ItmIndex==1) {t=3;}
            else { if (bcdmtp->ItmIndex==2) {t=4;}
                else {if (bcdmtp->ItmIndex==3) {t=5;}
                    else {ShowMessage("Введено не всі
інформацію");}
                }
            }
        }
        else {ShowMessage("Введено не всі інформацію");}
    }
    if (t!=0) {kol++;
        sfgT41->RgtKol=sfgT41->RgtKol+2;
        sfgT41->Stlbs[0]->Strks[2*kol]=bcmtpr->Text;
        sfgT41->Stlbs[0]->Strks[2*kol+1]="      "+bcdmtp->Text;
        Radz[kol]=Hntbf25[t-1][0];
        h[kol]=Hntbf25[t-1][1];
        float d;
        d=trg(eDiam->Text.c_str());
        float dk,Ky,er;
        er=36;
        if (t==1) { int i=0;
            bool f=false;
            while ((i<8)&&(!f))
            { if (d<=Hntbf16[i][0])
                {dk=Hntbf16[i][1];
                    f=true;}
                i++;
            }
            Ky=Hntbf29[1];
            float Td=Hntbf21[0];
            dlERT[0]=sqrt(dk*er*dk*er+0.25*0.25*Td*Td);
            dlERT[0]=rngt(dlERT[0],0);
            dlERT[kol]=Ky*dlERT[0];
        }
        else { if (t==3)
            { int i=0;
                bool f=false;
                while ((i<8)&&(!f))
                { if (d<=Hntbf16[i][0])
                    {dk=Hntbf16[i][2];
                        f=true;}
                    i++;
                }
                Ky=Hntbf29[4];
                dlERT[kol]=Ky*dk*er;
            }
        }
        T[kol]=Hntbf21[t];
        dlERT[kol]=rngt(dlERT[kol],0);
        float k=dlERT[kol-1]*dlERT[kol-1]+ERT[kol]*ERT[kol];
    }
}

```

```

        Zmn[kol]=2*(Radz[kol-1]+h[kol-1]+sqrt(k));
        Zxn[kol]=Zmn[kol]+T[kol-1];
        Zmx[kol]=Zxn[kol]+T[kol];
    }
}

void trmkL4::FormCreate(TObject *Sender)
{
    hnfL4->Width=320;
    hnfL4->Height=485;
    hnfL4->Left=396;
    pksL4->Width=310;
    pksL4->Height=451;
    tbinpt->TabVisible=true;
    tbinpt->Visible=true;
    tbmp->TabVisible=false;
    tbmp->Visible=false;
    tbkr42->TabVisible=false;
    tbkr42->Visible=false;
    sfgT41->Stlbs[0]->Strks[0]="План обробки";
    sfgT41->Stlbs[1]->Strks[0]="Rzi";
    sfgT41->Stlbs[2]->Strks[0]="hi";
    sfgT41->Stlbs[3]->Strks[0]="delEi";
    sfgT41->Stlbs[4]->Strks[0]="Ei";
    sfgT41->Stlbs[5]->Strks[0]="2Zimin,мкм";
    sfgT41->Stlbs[6]->Strks[0]="ДопускTdi,мкм";
    sfgT41->Stlbs[7]->Strks[0]="Zih,мкм";
    sfgT41->Stlbs[8]->Strks[0]="Zimax,мкм";
    sfgT41->Stlbs[9]->Strks[0]="di,мм";
    sfgT41->Stlbs[10]->Strks[0]="Вик.опер.розмір";
}

void trmkL4::bbtOKClick(TObject *Sender)
{
    if (kol!=0)
    {
        sfgT41->RgtKol=kol*2+2;
        sfgT41->FixedRows=1;
    }
    else {sfgT41->RgtKol=1;}
    hnfL4->Width=985;
    hnfL4->Height=569;
    hnfL4->Left=34;
    pksL4->Width=977;
    pksL4->Height=441;
    tbinpt->TabVisible=false;
    tbinpt->Visible=false;
    tbmp->TabVisible=true;
    tbmp->Visible=true;
    sfgT41->Stlbs[0]->Strks[1]="Заготовка";
    int i=0;
    bool f=false;
    float m=trg(eWeight->Text.c_str());
    float d=trg(eDiam->Text.c_str());
    fndts(Hntbf23b2,d,m);
    while ((i<6)&&(!f))
    {
        if (m<=Hntbf12[i][0])
        {
            f=true;
            Radz[0]=Hntbf12[i][1];
            h[0]=Hntbf12[i][2];
            T[0]=Hntbf21[0];
            Zmn[0]=0;
        }
        i++;
    }
    float k=d1ERT[0]*d1ERT[0]+ERT[1]*ERT[1];
    Zmn[1]=2*(Radz[0]+h[0]+sqrt(k));

```

```

Zxn[1]=Zmn[1]+fbs(crt2)*1000;
Zmx[1]=Zxn[1]+T[1];
for (i=0;i<=kol;i++)
{ sfgT41->Stlbs[1]->Strks[2*i+1]=(Antstr)Radz[i];
  sfgT41->Stlbs[2]->Strks[2*i+1]=(Antstr)h[i];
  sfgT41->Stlbs[6]->Strks[2*i+1]=(Antstr)T[i];
  sfgT41->Stlbs[3]->Strks[2*i+1]=(Antstr)dlERT[i];
  sfgT41->Stlbs[4]->Strks[2*i+1]=(Antstr)ERT[i];
  sfgT41->Stlbs[5]->Strks[2*i+1]=(Antstr)Zmn[i];
  sfgT41->Stlbs[7]->Strks[2*i+1]=(Antstr)Zxn[i];
  sfgT41->Stlbs[8]->Strks[2*i+1]=(Antstr)Zmx[i];
}
D[kol]=d*1000;
sfgT41->Stlbs[9]->Strks[2*kol+1]=(Antstr)d;
float qv1=trg(eDr1->Text.c_str());
float qv2=trg(eDr2->Text.c_str());
sfgT41->Stlbs[10]->Strks[2*kol+1]=sfgT41->Stlbs[9]->Strks[2*kol+1]+eDr1->Text;
sfgT41->Stlbs[10]->Strks[2*kol+1]=sfgT41->Stlbs[10]->Strks[2*kol+1]+eDr2->Text;
for (i=kol-1;i>=0;i--)
{ D[i]=D[i+1]+Zxn[i+1];
  float g=rngt(D[i]/1000,2);
  mshw2(sfgT41,9,2*i+1,g,2);
}
mshw2(sfgT41,10,1,rngt(D[0]/1000,1),1);
sfgT41->Stlbs[10]->Strks[1]=sfgT41->Stlbs[10]-
>Strks[1]+" "+(Antstr)(rngt(cs2,1)/10)+" "+(Antstr)(rngt(crt2,1)/10);
for (i=1;i<kol;i++)
{ float r=T[i];
  int q=0,o=1;
  while (r>1)
  { q++;
    o=o*10;
    r=r/10;
  }
  sfgT41->Stlbs[10]->Strks[2*i+1]=(Antstr)((rngt(D[i]/1000,q-1)/(o/10))+"
"+(Antstr)(-T[i]/1000);
}
sfgT41->Height=(sfgT41->RgtKol)*(sfgT41->Drtht)+sfgT41->GridLineWidth*(sfgT41-
>RgtKol+1)+10;
pksL4->Height=sfgT41->Height+170;
btNxt->Top=pksL4->Height-100;
hnfL4->Height=sfgT41->Height+220;
}
void trmkL4::FormShow(TObject *Sender)
{
hnfL4->Width=320;
hnfL4->Height=485;
hnfL4->Left=396;
pksL4->Width=310;
pksL4->Height=451;
tbinpt->TabVisible=true;
tbinpt->Visible=true;
tbmp->TabVisible=false;
tbmp->Visible=false;
tbkr42->TabVisible=false;
tbkr42->Visible=false;
}
void trmkL4::btNxtClick(TObject *Sender)
{
tbmp->TabVisible=false;
tbmp->Visible=false;
tbkr42->TabVisible=true;
tbkr42->Visible=true;
hnfL4->Width=685;

```

```

pksL4->Width=675;
if (kol>0)
    {sfgT42->RgtKol=sfgT41->RgtKol+1;}
    else {sfgT42->RgtKol=1;}
sfgT42->Stlbs[0]->Strks[sfgT42->RgtKol-1]="Підсумок";
float ZR1=0,ZR2=0;
for (int i=1;i<=kol;i++)
{ ZR1=ZR1+Zmn[i];
  ZR2=ZR2+Zxn[i];
}
sfgT42->Stlbs[1]->Strks[sfgT42->RgtKol-1]=(Antstr) ZR1;
sfgT42->Stlbs[2]->Strks[sfgT42->RgtKol-1]=(Antstr) ZR2;
sfgT42->Height=(sfgT41->RgtKol) * (sfgT41->Drtht) +50;
hnfL4->Height=(sfgT42->Height)+200;
pksL4->Height=(hnfL4->Height)-50;
bbtEnter->Top=(hnfL4->Height)-160;
btNxtLab->Top=(hnfL4->Height)-120;
sfgT42->Stlbs[0]->Strks[0]=sfgT41->Stlbs[0]->Strks[0];
sfgT42->Stlbs[1]->Strks[0]=sfgT41->Stlbs[5]->Strks[0];
sfgT42->Stlbs[2]->Strks[0]=sfgT41->Stlbs[7]->Strks[0];
sfgT42->Stlbs[3]->Strks[0]="Виконавчий розмір";
for (int i=1;i<=2*kol+1;i++)
{ sfgT42->Stlbs[0]->Strks[i]=sfgT41->Stlbs[0]->Strks[i];
  sfgT42->Stlbs[1]->Strks[i]=sfgT41->Stlbs[5]->Strks[i];
  sfgT42->Stlbs[2]->Strks[i]=sfgT41->Stlbs[7]->Strks[i];
  sfgT42->Stlbs[3]->Strks[i]=sfgT41->Stlbs[10]->Strks[i];
}
}
void trmkL4::bbtEnterClick(TObject *Sender)
{
kol=0;
hnfL4->Width=320;
hnfL4->Height=490;
hnfL4->Left=396;
pksL4->Width=310;
pksL4->Height=451;
tbinpt->TabVisible=true;
tbinpt->Visible=true;
tbmp->TabVisible=false;
tbmp->Visible=false;
tbkr42->TabVisible=false;
tbkr42->Visible=false;
}
void trmkL4::btNxtLabClick(TObject *Sender)
{
hnfL4->Close();
}
TrmkL5 *hnfL5;
struct pln5 {
    bool p[9];
    int psc[9];
};
struct inpdt5 {
    float A,B,D1,D2,D3;
    int TypeZag;
    float m1,m2;
    int Class;
    pln5 plnd;
};
inpdt5 FRGT;
float Hntbf52[9][6]={ {80,180,260,80,180,260},
                      {180,180,180,400,400,400},
                      {60,80,90,80,90,100},
                      {25,40,47,40,47,62}};

```

```

float Hntbf53[5][12]={0,0,0,50,50,120,120,260,0,120,120,260},
                    {120,260,1.4,1.0,1.4,1.2,2.1,1.5,0.4,0.04,0.5,0.06}};

typedef float Hntbf23[18][28];
typedef float es[4];
es cs;
Hntbf23
trmkL5::TrmkL5(TComponent* Vldc)
    : TForm(Vldc)
{
    sfgT51->Stlbs[0]->Strks[0]="Номер та зміст переходу";
    sfgT51->Stlbs[1]->Strks[0]="Забезпечувальний розмір";
    sfgT51->Stlbs[2]->Strks[0]="Припуск Zimin, мм";
    sfgT51->Stlbs[3]->Strks[0]="Припуск Zimax, мм";
    sfgT51->Stlbs[4]->Strks[0]="Технологічний допуск, мм";
    srgtm ->Stlbs[0]->Strks[0]="Операція";
    srgtm ->Stlbs[1]->Strks[0]="Розмір";
    srgtm ->Stlbs[2]->Strks[0]="Величина, мм";
    srgtm ->Stlbs[3]->Strks[0]="Технологічний допуск, мм";
    sgnfm->Stlbs[0]->Strks[0]="Розмір";
    sgnfm->Stlbs[1]->Strks[0]="Величина, мм";
}
void trmkL5::FormClose(TObject *Sender, TCloseAction &Action)
{
    ftmvyn->FormCreate(hnfl5);
}
float mrnd(float n,int q){
    for (int i=0;i<q;i++)
    { n=n*10;
    }
    float nf1=floor(n*10);
    n=floor(n);
    float nf2=n*10;
    nf2=nf1-nf2;
    if (nf2>=5) {n=n+1;}
    return n;
}
void trmkL5::vbyrChange(TObject *Sender)
{
    cdbls->Text="";
    cdbls->Items->Clear();
    if (vbyr->ItmIndex==0)
    {cdbls->Items->Add("1");
    cdbls->Items->Add("2");
    cdbls->Items->Add("3");
    }

    if (vbyr->ItmIndex==1)
    {cdbls->Items->Add("Підвищений");
    cdbls->Items->Add("Нормальний");
    }
}
float mfnd2(float b,float t,int tp){
    int i=0;
    bool f=false;
    while ((i<6)&&(!f))
    { if ((b<=Hntbf52[0][i])&&(t<=Hntbf52[1][i]))
        {float rzlt=Hntbf52[tp][i]/1000;
        rzlt=mrnd(rzlt,3);
        return rzlt;
        f=true;
        }
    i++;
}
}

```

```

}
void mshw(TStringGrid* sg,int c1,int c2,float z,int zr){
    float Zr1=z;
    for (int i=0;i<zr;i++)
    { Zr1=Zr1/10;
    }
    Zr1=floor(Zr1);
    float Zr2=z,cZr2=Zr1;
    for (int i=0;i<zr;i++)
    { cZr2=cZr2*10;}
    Zr2=Zr2-cZr2;
    Antstr s;
    int t=1;
    for (int i=0;i<zr-1;i++)
    {t=t*10;}
    if (Zr2>=t)
        {s=(Antstr)Zr1+"."+(Antstr)Zr2;}
    else { s=(Antstr)Zr1+"."+ "0";
        t=t/10;
        while (Zr2<t)
        { s=s+"0";
            t=t/10;
        }
        s=s+(Antstr)Zr2;
    }
    sg->Stlbs[c1]->Strks[c2]=s;
}
void fndts23(Hntbf23 hntbf){
    for (int i=1;i<14;i++)
    { if ((FRGT.A>=hntbf[0][i*2])&&(FRGT.A<hntbf[0][i*2+1]))
        { for (int j=1;j<18;j++)
            { if ((FRGT.m1>=hntbf[j][0])&&(FRGT.m2<=hntbf[j][1]))
                {cs[0]=hntbf[j][i*2];
                cs[1]=hntbf[j][i*2+1];
                }
            }
        }
    }
    for (int i=1;i<14;i++)
    { if ((FRGT.B>=hntbf[0][i*2])&&(FRGT.B<hntbf[0][i*2+1]))
        { for (int j=1;j<18;j++)
            { if ((FRGT.m1>=hntbf[j][0])&&(FRGT.m2<=hntbf[j][1]))
                {cs[2]=hntbf[j][i*2];
                cs[3]=hntbf[j][i*2+1];
                }
            }
        }
    }
}
void trmkL5::btnPln5OKClick(TObject *Sender)
{
    FRGT.D1=trg(eDr1->Text.c_str());
    FRGT.D2=trg(eDr2->Text.c_str());
    FRGT.D3=trg(eD3->Text.c_str());
    FRGT.A=trg(eA->Text.c_str());
    FRGT.B=trg(eB->Text.c_str());
    FRGT.m1=trg(eMore->Text.c_str());
    FRGT.m2=trg(eLess->Text.c_str());
    FRGT.TypeZag=vbyr->ItmIndex;
    FRGT.Class=cdbls->ItmIndex+1;
    FRGT.plnd.p[0]=cdb1->Checked;
    FRGT.plnd.p[1]=cdb2->Checked;
    FRGT.plnd.p[2]=cdb3->Checked;
    FRGT.plnd.p[3]=cdb4->Checked;
}

```

```

FRGT.plnd.p[4]=cdb5->Checked;
FRGT.plnd.p[5]=cdb6->Checked;
FRGT.plnd.p[6]=cdb7->Checked;
FRGT.plnd.p[7]=cdb8->Checked;
FRGT.plnd.p[8]=cdb9->Checked;
if (vbyr->ItmIndex==1)
{
    if (cdbls->ItmIndex==0)
        {fndts3(Hntbf23a);}
        else {fndts3(Hntbf23b);}
}
if (vbyr->ItmIndex==0)
{
    fndts3(cdbls->ItmIndex+1,FRGT.A,0);
    fndts3(cdbls->ItmIndex+1,FRGT.B,2);
}
float esA=cs[0],eiA=cs[1],esB=cs[2],eiB=cs[3];
float esAzag=0.5*(esA+esB),eiAzag=0.5*(eiA+eiB);
int kol=0,kol2=0,kolA=0,kolB=0;
if (FRGT.plnd.p[0]) {kolB++;}
if (FRGT.plnd.p[1]) {kolB++;}
if (FRGT.plnd.p[2]) {kolB++;}
if (FRGT.plnd.p[3]) {kolB++;}
if (FRGT.plnd.p[4]) {kolA++;}
if (FRGT.plnd.p[5]) {kolA++;}
if (FRGT.plnd.p[6]) {kolB++;}
if (FRGT.plnd.p[7]) {kolB++;}
if (FRGT.plnd.p[8]) {kolA++;}
for (int i=0;i<9;i++)
{FRGT.plnd.psc[i]=0;}
if (FRGT.plnd.p[0]) {kol++;kol2++;FRGT.plnd.psc[0]=kol;
    sfgT51->Stlbs[0]->Strks[kol2]=(Antstr) kol+" ";
    sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr) kol;
    kolB=kol;
}
if (FRGT.plnd.p[2]) {kol++;kol2++;FRGT.plnd.psc[2]=kol;
    sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr) kol;
    kolB=kol;
    sfgT51->Stlbs[0]->Strks[kol2]=(Antstr) kol+" ";}
if (FRGT.plnd.p[4]) {if (!FRGT.plnd.p[2])
    {kol++;
        sfgT51->Stlbs[0]->Strks[kol2+1]=(Antstr) kol+" ";
    }
    FRGT.plnd.psc[4]=kol;
    kol2++;
    sfgT51->Stlbs[1]->Strks[kol2]="A"+(Antstr) kol;
    kolA=kol;
}
if (FRGT.plnd.p[1]) {kol++;kol2++;FRGT.plnd.psc[1]=kol;
    sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr) kol;
    sfgT51->Stlbs[0]->Strks[kol2]=(Antstr) kol+" ";
    kolB=kol;
}
if (FRGT.plnd.p[3]) {kol++;kol2++;FRGT.plnd.psc[3]=kol;
    sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr) kol;
    sfgT51->Stlbs[0]->Strks[kol2]=(Antstr) kol+" ";
    kolB=kol;
}
if (FRGT.plnd.p[5]) {if (!FRGT.plnd.p[3])
    {kol++;
        sfgT51->Stlbs[0]->Strks[kol2+1]=(Antstr) kol+" ";
    }
    FRGT.plnd.psc[5]=kol;
    kol2++;
    sfgT51->Stlbs[1]->Strks[kol2]="A"+(Antstr) kol;
    kolA=kol;
}

```

```

    }
    if (FRGT.plnd.p[6]) {kol++;kol2++;FRGT.plnd.psc[6]=kol;
        sfgT51->Stlbs[0]->Strks[kol2]=(Antstr)kol+". ";
        sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr)kol;
        kolB=kol;
    }
    if (FRGT.plnd.p[7]) {kol++;kol2++;FRGT.plnd.psc[7]=kol;
        kolB=kol;
        sfgT51->Stlbs[0]->Strks[kol2]=(Antstr)kol+". ";
        sfgT51->Stlbs[1]->Strks[kol2]="B"+(Antstr)kol;
    }
    if (FRGT.plnd.p[8]) {kol++;kol2++;FRGT.plnd.psc[8]=kol;
        kolA=kol;
        sfgT51->Stlbs[0]->Strks[kol2]=(Antstr)kol+". ";
        sfgT51->Stlbs[1]->Strks[kol2]="A"+(Antstr)kol;
    }
    sfgT51->RgtKol=kol2+1;
    hnfl5->srgtm ->RgtKol=kol2+1;
    for (int i=1;i<=kol2;i++)
    { hnfl5->srgtm ->Stlbs[1]->Strks[i]=sfgT51->Stlbs[1]->Strks[i];
    }
    kol2=0;
    hnfl5->Width=1010;
    pksL5->Width=1000;
    hnfl5->Left=10;
    float Z[9],T[9],Zm[9],A[9],B[9];
    for (int i=0;i<9;i++)
    {Z[i]=0;T[i]=0;Zm[i]=0;A[i]=0;B[i]=0;}
    if (FRGT.plnd.p[0]) {Z[kol2]=myfind(FRGT.B,FRGT.D1,1,0);
        T[kol2]=mfnd2(FRGT.B,FRGT.D1,2);
        mshw(sfgT51,2,kol2+1,Z[kol2],2);
        mshw(sfgT51,4,kol2+1,T[kol2],3);
        kol2++;
        sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb1->Caption;
        srgtm ->Stlbs[0]->Strks[kol2]=cdb1->Caption;
        if (FRGT.plnd.psc[0]==kol)
            {Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
            else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
        mshw(sfgT51,3,kol2,Zm[kol2-1],2);
        B[FRGT.plnd.psc[0]]=Zm[kol2-1];
        if (A[FRGT.plnd.psc[0]]==0) {A[FRGT.plnd.psc[0]]=Zm[kol2-
1];}
    }
    if (FRGT.plnd.p[2]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb3->Caption;
        srgtm ->Stlbs[0]->Strks[kol2]=cdb3->Caption;
        Z[kol2-1]=myfind(FRGT.B,FRGT.D2,1,0);
        T[kol2-1]=mfnd2(FRGT.B,FRGT.D2,2);
        mshw(sfgT51,2,kol2,Z[kol2-1],2);
        mshw(sfgT51,4,kol2,T[kol2-1],3);
        if (FRGT.plnd.psc[2]==kol)
            {Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-1]/1000,2);}
            else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
        mshw(sfgT51,3,kol2,Zm[kol2-1],2);
        B[FRGT.plnd.psc[2]]=Zm[kol2-1];
        if (A[FRGT.plnd.psc[2]]==0) {A[FRGT.plnd.psc[2]]=Zm[kol2-
1];}
    }
    if (FRGT.plnd.p[4]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb5->Caption;
        srgtm ->Stlbs[0]->Strks[kol2]=cdb5->Caption;
        Z[kol2-1]=myfind(FRGT.A,FRGT.D1,1,0);
    }

```

```

T[kol2-1]=mfnd2(FRGT.A,FRGT.D1,2);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[4]==kol)
{Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiAzag),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
A[FRGT.plnd.psc[4]]=Zm[kol2-1];
if (B[FRGT.plnd.psc[4]]==0) {B[FRGT.plnd.psc[4]]=Zm[kol2-
1];}

}

if (FRGT.plnd.p[1]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb2->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb2->Caption;
Z[kol2-1]=myfind(FRGT.B,FRGT.D1,1,1);
T[kol2-1]=mfnd2(FRGT.B,FRGT.D1,4);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[1]==kol)
{Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
B[FRGT.plnd.psc[1]]=Zm[kol2-1];
if (A[FRGT.plnd.psc[1]]==0) {A[FRGT.plnd.psc[1]]=Zm[kol2-
1];}

}

if (FRGT.plnd.p[3]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb4->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb4->Caption;
Z[kol2-1]=myfind(FRGT.B,FRGT.D2,1,1);
T[kol2-1]=mfnd2(FRGT.B,FRGT.D2,4);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[3]==kol)
{Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-1]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
B[FRGT.plnd.psc[3]]=Zm[kol2-1];
if (A[FRGT.plnd.psc[3]]==0) {A[FRGT.plnd.psc[3]]=Zm[kol2-
1];}

}

if (FRGT.plnd.p[5]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb6->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb6->Caption;
Z[kol2-1]=myfind(FRGT.A,FRGT.D1,1,1);
T[kol2-1]=mfnd2(FRGT.A,FRGT.D1,4);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[5]==kol)
{Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiAzag),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
A[FRGT.plnd.psc[5]]=Zm[kol2-1];
if (B[FRGT.plnd.psc[5]]==0) {B[FRGT.plnd.psc[5]]=Zm[kol2-
1];}

}

if (FRGT.plnd.p[6]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb7->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb7->Caption;
Z[kol2-1]=myfind(FRGT.B,FRGT.D1,0,0);

```

```

T[kol2-1]=mfnd2(FRGT.B,FRGT.D1,6);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[6]==kol)
    {Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
B[FRGT.plnd.psc[6]]=Zm[kol2-1];
if (A[FRGT.plnd.psc[6]]==0) {A[FRGT.plnd.psc[6]]=Zm[kol2-
1];}

}
if (FRGT.plnd.p[7]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb8->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb8->Caption;
Z[kol2-1]=myfind(FRGT.B,FRGT.D2,0,0);
T[kol2-1]=mfnd2(FRGT.B,FRGT.D2,7);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[7]==kol)
    {Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiB),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
B[FRGT.plnd.psc[7]]=Zm[kol2-1];
if (A[FRGT.plnd.psc[7]]==0) {A[FRGT.plnd.psc[7]]=Zm[kol2-
1];}

}
if (FRGT.plnd.p[8]) {kol2++;sfgT51->Stlbs[0]->Strks[kol2]=sfgT51->Stlbs[0]-
>Strks[kol2]+cdb9->Caption;
srgtm ->Stlbs[0]->Strks[kol2]=cdb9->Caption;
Z[kol2-1]=myfind(FRGT.A,FRGT.D1,0,1);
T[kol2-1]=mfnd2(FRGT.A,FRGT.D1,8);
mshw(sfgT51,2,kol2,Z[kol2-1],2);
mshw(sfgT51,4,kol2,T[kol2-1],3);
if (FRGT.plnd.psc[8]==kol)
    {Zm[kol2-1]=mrnd(Z[kol2-1]/100+T[kol2-2]/1000,2);}
else {Zm[kol2-1]=mrnd(Z[kol2-1]/100+0.5*fbs(eiAzag),2);}
mshw(sfgT51,3,kol2,Zm[kol2-1],2);
A[FRGT.plnd.psc[8]]=Zm[kol2-1];
if (B[FRGT.plnd.psc[8]]==0) {B[FRGT.plnd.psc[8]]=Zm[kol2-
1];}

}
for (int i=1;i<=kol2;i++)
{ hnfL5->srgtm ->Stlbs[3]->Strks[i]=sfgT51->Stlbs[4]->Strks[i];
}
for (int i=0;i<kol;i++)
{ if (i>kolA)
    {A[i]=0;}
else {A[i]=A[i+1];}
if (i>kolB)
    {B[i]=0;}
else {B[i]=B[i+1];}
}
A[kolA]=FRGT.A*100;
B[kolB]=FRGT.B*100;
for (int i=kolA-1;i>=0;i--)
{ A[i]=A[i+1]+A[i];}
for (int i=kolB-1;i>=0;i--)
{ B[i]=B[i+1]+B[i];}
for (int i=1;i<=kol2;i++)
{ Antstr s=srgtm ->Stlbs[1]->Strks[i];
int q=s[2]-'0';

```

```

    if (s[1]=='B')
        { mshw(srgtm ,2,i,B[q],2);}
        else { mshw(srgtm ,2,i,A[q],2);}
}
sgnfm->RgtKol=kolA+kolB+1;
sgnfm->Height=sgnfm->Drtht*sgnfm->RgtKol+10;
for (int i=kolA-1;i>=0;i--)
{ sgnfm->Stlbs[0]->Strks[i+1]="A"+(Antstr)i;
  mshw(sgnfm,1,i+1,A[i],2);
}
for (int i=kolB-1;i>=0;i--)
{ sgnfm->Stlbs[0]->Strks[i+kolA+1]="B"+(Antstr)i;
  mshw(sgnfm,1,i+kolA+1,B[i],2);
}
sfgT51->Height=sfgT51->Drtht*(kol2+1)+10;
srgtm ->Height=srgtm ->Drtht*(kol2+1)+10;
tbinpt->TabVisible=false;
tbkr51->TabVisible=true;

```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**Дослідження та програмна реалізація
засобів розрахунку параметрів
механічної обробки деталей**

Виконав: ст. гр. КНТ-122м

Каріна ЗАКІНА

Керівник:
к. т. н., доцент

Тетяна КОЛПАКОВА

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для механічного оброблення деталей.

Предмет дослідження – програмні засоби для механічного оброблення деталей.

Метою роботи є дослідження та розробка програмного забезпечення для розрахунку параметрів механічної обробки деталей.

2

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення мети роботи потрібно розв'язати такі завдання:

- виконати аналіз предметної області та програмних засобів для розрахунку параметрів механічної обробки деталей;
- розробити модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей;
- здійснити проектування програмного забезпечення для розрахунку параметрів механічної обробки деталей;
- створити програмне забезпечення для розрахунку параметрів механічної обробки деталей;
- виконати тестування розробленого програмного забезпечення для розрахунку параметрів механічної обробки деталей.

3

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для підтримки процесу механічної обробки деталей

Критерій порівняння	Fusion 360	Abaqus	Caesar II
Засоби інтегрованого середовища розробки	–	+	–
Безкоштовність	+–	–	–
Засоби фізичного аналізу	–	+	+–
Висока якість візуалізації	+	+–	+–
Підтримка різних платформ	+–	–	–

4

Рисунок В.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	Ruby
Зручність для розробки програмного забезпечення для розрахунку параметрів механічної обробки деталей	+	–	–
Безпека даних	+	+	+-
Робота з пам'яттю	+	–	–
Велика спільнота розробників	+	+	+-
Розширюваність	+	+	+-

5

Рисунок В.5 – Слайд 5

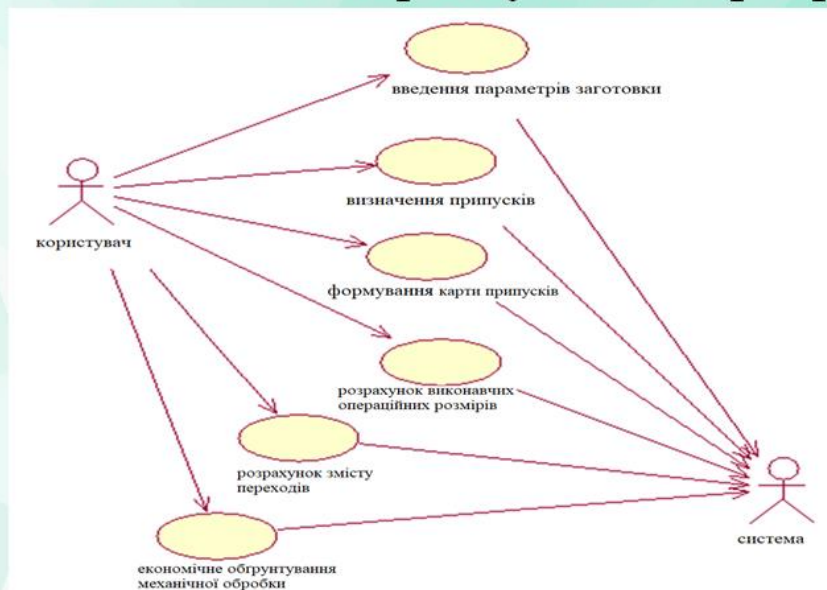
Порівняння середовищ розробки ПЗ

Критерій порівняння середовищ розробки	Середовище розробки		
	C++Builder	Visual Studio Code	Eclipse
Компоненти та бібліотеки	+	+-	+
Мови програмування	C++	Різні мови	Різні мови
Зручність редактору коду	+	+-	+
Безкоштовність	+-	+	+
Зручні інструменти для розробки на C++	+	+	–

6

Рисунок В.6 – Слайд 6

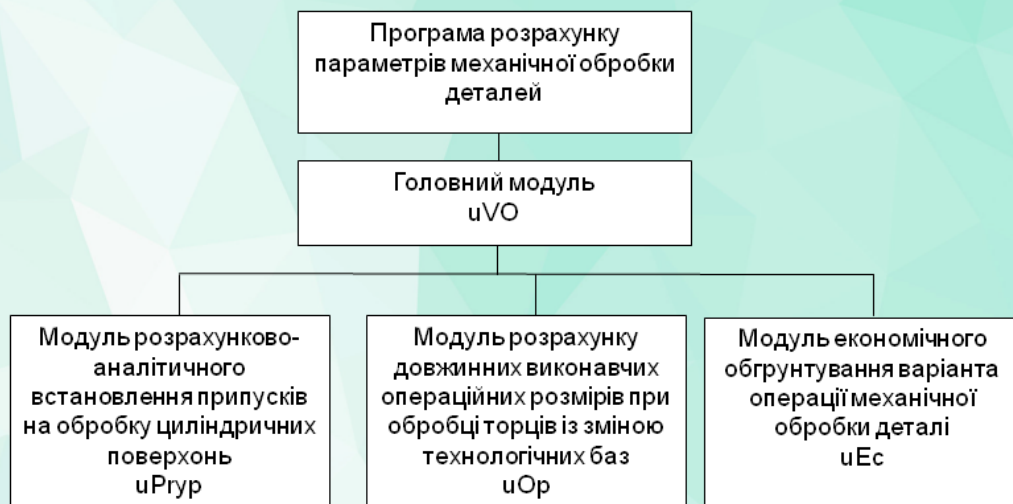
Модель взаємодії користувача з програмою



7

Рисунок В.7 – Слайд 7

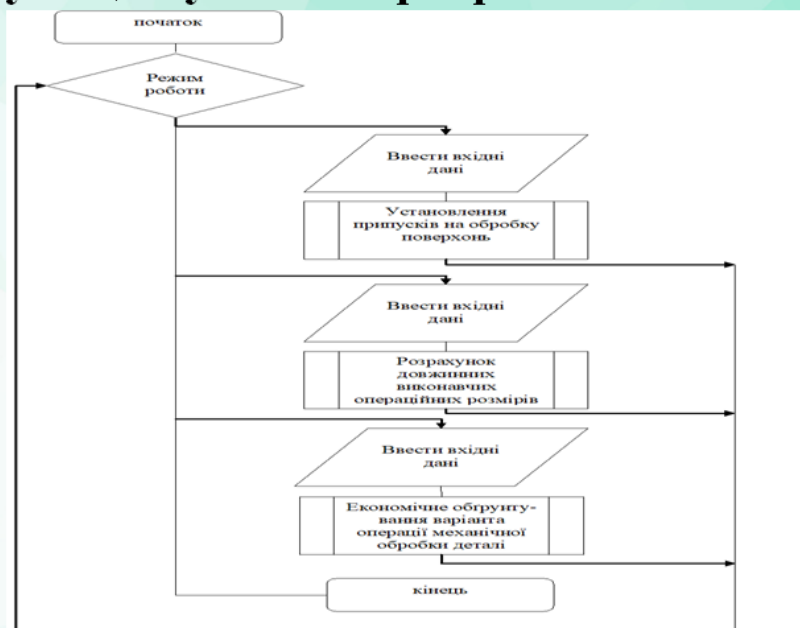
Структура програмного забезпечення



8

Рисунок В.8 – Слайд 8

Схема функціонування програмного забезпечення



9

Рисунок В.9 – Слайд 9

Робота з програмою. Розрахунок параметрів механічної обробки деталей

The image displays two screenshots of the 'VO 1.0' software interface, which is used for calculating mechanical processing parameters. The interface is divided into two main sections: 'Заготовка' (Workpiece) and 'План обробки поверхні' (Surface processing plan).

Left Screenshot (Initial State):

- Заготовка:** Fields for 'Маса' (Mass), 'Діаметр' (Diameter), 'Допуски' (Tolerances), and 'Тип поверхні' (Surface type) are empty.
- План обробки поверхні:** Fields for 'Тип обробки' (Processing type) and 'Спосіб обробки' (Processing method) are empty.
- Buttons: 'Додати' (Add) and 'Розрахувати' (Calculate).

Right Screenshot (Filled State):

- Заготовка:** 'Маса' is 3.8, 'Діаметр' is 35, 'Допуски' are +0.05 and +0.02, and 'Тип поверхні' is 'Вал' (Shaft).
- План обробки поверхні:** 'Тип обробки' is 'Шліфування' (Grinding) and 'Спосіб обробки' is 'Чистове' (Finish).
- Buttons: 'Додати' (Add) and 'Розрахувати' (Calculate).

10

Рисунок В.10 – Слайд 10

Робота з програмою. Карта розрахунку припусків

VO 1.0
Карта розрахунку припусків

План обробки	Ra	Rz	deEI	EI	ZZмінен	Допуст.Т.бізн	Змінен	Zмакс.зн	бізн	Вик. опер. розмір
Заготовка	160	200	500	0	0	2000	0	0	38.43	38.4 +1.3 -0.7
Обґранування										
Чорнове	50	50	30	0	1720	390	2420	2810	36.01	36.01 -0.39
Обґранування										
Чистове	25	25	0	0	260	160	650	810	35.36	35.36 -0.16
Шліфування										
Чорнове	10	20	1	0	100	39	260	299	35.10	35.1 -0.039
Шліфування										
Чистове	5	15	0	0	62	30	101	131	35	35+0.05+0.02

Далі

11

Рисунок В.11 – Слайд 11

Робота з програмою. Розрахунок виконавчих операційних розмірів

VO 1.0
Вхідні дані

Розмір деталі

D1

D2

D3

A

B

Заготовка

Тип

Маса: більш до

Клас точності

План обробки торця

☐ Чорнова підрізка торця M

☐ Чистова підрізка торця M

☐ Чорнова підрізка торця P

☐ Чистова підрізка торця P

☐ Чорнова підрізка торця N

☐ Чистова підрізка торця N

☐ Попереднє шліфування торця M

☐ Одноратне шліфування торця P

☐ Чистове шліфування торця N

Генерувати

VO 1.0
Вихідні операції

Вимір та зміст параметру	Вибірочний розмір	Припуск Zmax, мм	Припуск Zmin, мм	Технологічний допуск, мм
1. Чорнова підрізка торця M	B1	1.20	1.55	0.530
2. Чорнова підрізка торця P	B2	1.20	1.55	0.530
Чистова підрізка торця N	A2	0.90	1.25	0.480
3. Чистова підрізка торця N	A3	0.70	1.10	0.120

Далі

12

Рисунок В.12 – Слайд 12

Робота з програмою. Розрахунок економічного обґрунтування варіанта операції

The screenshot shows the 'VO 1.0' software interface. The left panel contains input fields for various parameters, and the right panel displays a table of calculated values.

Вхідні дані (Input Data):

- Річна програма випуску: 100 x 10³
- Параметри зубчатого колеса:
 - m: 3 мм
 - z: 30 мм
 - B: 30 мм
- Діаметр фрези: 112
- Розряд робітника: 4
- Модель зубофрезерного напівавтомата: 5306
- Модель зубодобального напівавтомата: 5121

Додаткові дані (Additional Data):

- Відсоток перевищення норми: 120
- Тип виробництва: Массовое
- Нормативний коефіцієнт ефективності: 0.15
- Вартість 1 кВт/г електроенергії: 1.32
- Дійсний річний фонд праці обладнання: 4015
- Відсотки вправдань від балансової вартості обладнання:
 - на амортизацію: 15
 - на обслуговування: 0.5
 - на ремонт обладнання: 4.5

Таблиця розрахунків (Calculation Table):

Статті/Режим	Зубофрезерування	Зубодобання
Основний час на операцію	6.28	3.35
Норма штучного часу	7.02	3.79
Кількість верстатів	3	2
Верстатосмісткість	6.85	3.16
Витрати на:		
- зарплатну плату	1.26	1.26
- електроенергію	0.08	0.04
- амортизацію обладнання	0.38	0.43
- обслуговування обладнання	0.01	0.01
- ремонт обладнання	0.11	0.13
- інструмент	0.245	0.590
Технологічна собівартість операції	0.1220	0.0778
Економічна доцільність	-	+

Результати:

- Економічний ефект: 4999.19
- Звіт про вхідні дані
- Обрати дані роботи

13

Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для механічного оброблення деталей.

Визначено функціональні вимоги до розробки програмного забезпечення. Для створення програми було обрано мову програмування C++ та середовище розробки C++Builder.

Наукова новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для розрахунку параметрів механічної обробки деталей, яка подана у вигляді діаграми прецедентів у нотації UML, та дозволяє розроблювати програмне забезпечення для розрахунку параметрів механічної обробки деталей.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення для розрахунку параметрів механічної обробки деталей.

Результати тестування програми показали, що вона працює коректно та може використовуватися за призначенням.

14

Рисунок В.14 – Слайд 14