

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Інформаційне та програмне забезпечення системи
розрахунку колінчатого вала»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Шелехов І.В.

Студента групи ІН 73 – 9

Шилов К. М.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 г.

**ЗАВДАННЯ
до випускної роботи**

Студента четвертого курсу, групи ІН-73-9 спеціальності “Комп'ютерні науки”
денної форми навчання Шилова Костянтина Миколайовича.

**Тема: “Інформаційне та програмне забезпечення системи розрахунку
колінчатого вала ”**

Затверджена наказом по СумДУ

№ _____ от _____ 2021 г.

Зміст пояснювальної записки: 1)Дослідження галузі машинобудування; 2) аналітичний огляд програмних комплексів для розрахунку колінчатих валів; 3) постановка завдання й формування завдань дослідження; 4)опис засобів для програмної реалізації; 5)розробка інформаційного та програмного забезпечення системи розрахунку колінчатого вала; 6)тестування роботи розробленого продукту та аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 г.

Керівник випускної роботи _____ Шелехов І.В.

Завдання прийняв до виконання _____ Шилов К. М.

РЕФЕРАТ

Записка: 49 стор., 24 рис., 2 табл., 2 додатки, 10 бібліографічних найменувань.

Об'єкт дослідження – процес проектування системи розрахунку колінчатого вала.

Мета роботи – розробка інформаційного та програмного забезпечення системи розрахунку колінчатого вала.

Методи дослідження – методи проектування інформаційних систем, методи тестування програмного забезпечення.

Результати – Дипломний проект присвячений розробці інформаційного та програмного забезпечення системи розрахунку колінчатого вала. Під час здійснення дипломного проектування було досліджено галузь машинобудування, визначено актуальність роботи. Також було проведено аналіз аналогів, який показав, що існує декілька програмних комплексів, але вони занадто дорогі та громіздкі. Були сформульовані мета та задачі. Також було створено його блок-схему та розроблено структуру додатку. Наступним кроком було створення самого додатку для розрахунку колінчатого вала. Потім відбулося його тестування. Було здійснено навмисне введення недопустимих даних або їх видалення з поля вводу. Додаток відреагував на це відповідними повідомленнями. Також було перевірено можливість збереження результату при порожньому полі виводу (додаток реагує правильно – видає відповідне повідомлення). Результатом проведеної роботи є протестований розрахунковий додаток.

КОЛІНЧАТИЙ ВАЛ, КРИВОШИП, ШКІВ, MSC NASTRAN, ANSYS,
KVAL, MS EXCEL, C++, PYTHON, C#, JAVASCRIPT, VISUAL BASIC,
C++ BUILDER

ЗМІСТ

Вступ.....	5
1 Дослідження предметної галузі	6
1.1 Дослідження актуальності проблеми	10
1.2 Аналіз аналогів	12
1.2.1 MSC Nastran	12
1.2.2 ANSYS Mechanical Enterprise.....	13
1.2.3 KVAL	16
1.3 Постановка задачі.....	17
1.3.1 Мета задачі дослідження	17
2 Вибір засобів реалізації	18
3 Практичний розділ	22
3.1 Визначення структури додатку.....	22
3.2 Розробка дизайну додатку	23
3.3 Розробка функціоналу додатку	25
3.4 Тестування програмного додатку	29
Висновок.....	31
Список використаної літератури	32
Додаток А.....	33
Додаток Б.....	34

ВСТУП

У даний час відбувається всесвітній процес автоматизації та комп'ютеризації всіх сфер людської діяльності.

Автоматизація процесів, що відбуваються на підприємстві, є показником професійного та серйозного підходу до оптимізації діяльності організації. Рішення керівника про автоматизацію систем розрахунку можна розглядати як крок до підвищення ефективності діяльності підприємства.

Машинобудування є однією з галузей, чий проекти автоматизації є досить актуальними для більшої частини підприємств. Автоматизації сьогодні піддається планування, облік матеріальних і товарних цінностей, безпосереднє управління виробництвом і багато інших внутрішніх бізнес-процесів, які є характерними для машинобудівних підприємств.

На даний момент абсолютно зрозуміло, що автоматизації потребують не тільки маркетинг або корпоративне керівництво, а й інші сфери діяльності підприємств. Застосування автоматизації для складання графіків експлуатації обладнання й пересувного складу, складських приміщень і заводських цехів лягає в основу виробничої логістики.

Автоматизація та інформаційні технології (ІТ) є необхідними як на стадії проектування і виробництва, так і на стадії реалізації готової продукції. Можливості, які надає застосування ІТ при проектуванні в машинобудуванні є найефективнішими. Розробка й оптимізація спеціалізованих програмних засобів, які дозволяють швидко розрахувати та в 3D форматі представити будь-яку деталь, агрегат, не просто на зображенні, але і в дії, відкриває перед проектувальниками широкий спектр можливостей. Те, на що раніше йшли роки ретельної праці, сьогодні стає доступним за кілька хвилин.

Якісне проектування та впровадження системи є основною передумовою її ефективного функціонування при постійному вдосконаленні. Більш того, зацікавленим користувачам набагато простіше й зручніше отримати результати розрахунку в електронній формі, ніж у рукописному варіанті.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ

Колінчатий вал – вал (чи вузол у випадку складеного вала) складної форми, призначений для перетворення зворотно-поступального руху (наприклад, поршня) в обертальний навколо власної осі, що має шийки, зміщені від осі обертання для кріплення шатунів, від яких приймає зусилля та перетворює їх в обертальний момент (рис. 1.1).



Рисунок 1.1 – Колінчатий вал

Дана деталь є складовою частиною кривошипно-шатунного механізму (КШМ)(рис. 1.2).



Рисунок 1.2 – Кривошипно-шатунний механізм

Колінчатий вал використовується в двигунах внутрішнього згорання, поршневих компресорах, насосах, ковальсько-пресових машинах та ін.

Основні елементи колінчатого вала є наступними:

- корінна шийка – опора вала, що лежить в корінному підшипнику, розміщеному в картері (металевий корпус) двигуна;
- шатунна шийка – опора, за допомогою якої вал зв'язується з шатунами (для змазування шатунних підшипників є масляні канали);
- щоки – зв'язують корінні та шатунні шийки;
- вихідна частина вала – частина вала, на якій кріпиться зубчасте колесо або шків (колесо з обідком, або з канавкою по ободу, яке передає чи приймає рух і зусилля приводному пасу) відбору потужності (рис. 1.3) для приводу кулачкового вала газорозподільного механізму і різних допоміжних вузлів, систем і агрегатів;
- задня вихідна частина вала (хвостовик) – частина вала що з'єднується з маховиком або шестернею відбору основної частини потужності;
- противаги – забезпечують розвантаження корінних підшипників від відцентрових сил інерції першого порядку незрівноважених мас кривошипа і нижньої частини шатуна, що забезпечує плавність роботи КШМ.



Рисунок 1.3 – Шків відбору потужності

Завдання розробки й оптимізації деталей двигуна, зокрема колінчатого вала як однієї з найбільш напружених і відповідальних деталей є складним і вимагає всебічного використання математичного моделювання.

Недостатня надійність колінчатого вала, як правило, спричиняє підвищені зноси та скорочення терміну служби двигуна. Від міцності колінчатого вала залежить можливість форсування двигуна.

Поломки колінчатих валів в експлуатації показують, що завдання підвищення міцності вала залишається актуальним, особливо при постійному форсуванні двигунів (в першу чергу, дизелів) за допомогою нагнітання повітря.

Аналіз поломок колінчатих валів показує, що поломки здебільшого виникають через зношення деталі. Руйнування починається від зон найбільших концентрацій напруг, розташованих біля країв отворів для змазування в шийках або (частіше) в галтелях (форма поверхні у вигляді жолобка, виїмки на зовнішньому або внутрішньому ребрі деталі) сполучення щік із шийками.

При необроблених внутрішніх поверхнях порожнистих шийок руйнування іноді починаються з внутрішніх порожнин, незважаючи на більш низький рівень номінальних напруг в порівнянні з зовнішньою поверхнею.

Розрахунок вала на витривалість пов'язаний із визначенням запасів міцності його елементів. Витривалість у значній мірі залежить від величини й характеру діючих навантажень, форми та розмірів елементів колінчатого вала, структури матеріалу, способів обробки, форми переходів і сполучень, асиметрії циклу та ін..

Колінчатий вал представляє просторову, статично невизначену систему на пружних опорах із різною жорсткістю окремих ділянок. Незважаючи на досягнутий прогрес в області обчислювальної техніки та розробки прикладних розрахункових програм, загальний чисельний розрахунок складного колінчатого вала багатопілого двигуна з урахуванням всього спектру навантажень і граничних умов, із урахуванням впливу сполучених із валом елементів підшипникових опор, є надзвичайно трудомістким завданням.

У даній ситуації більш доцільним є використання ієрархічного підходу чисельного розрахунку, прийнятого в інженерній практиці розрахунків базових деталей двигуна внутрішнього згоряння (ДВЗ) із використанням як балкових, так і об'ємних моделей. При такому підході різні етапи спільної задачі раціонально виконувати із застосуванням математичних моделей різного рівня.

Для визначення коефіцієнтів концентрації напружень в основному використовуються емпіричні залежності, де коефіцієнти є функцією різних безрозмірних геометричних співвідношень елементів вала. Із розвитком двигунів внутрішнього згоряння розширився діапазон зміни геометричних параметрів вала, які впливають на концентрацію напружень. Так колінчаті вали сучасних форсованих двигунів виробляються зі значним перекриттям шийок, які виходять за межі застосованості отриманих раніше емпіричних залежностей для оцінки коефіцієнтів концентрації напружень. Приведена вище інформація стосується відносин діаметрів шийок до товщини щік, перекриття шийок до їх діаметру, наявності косих масло підвідних отворів та ін.. У цьому випадку альтернативою є кінцево-елементний розрахунок коефіцієнтів концентрації напружень. Для таких розрахунків використовуються об'ємні кінцево-елементні моделі з точним описом геометрії зон концентрації.

У загальній картині динамічної завантаженості колінчатих валів велику роль відіграють коливальні процеси. Додаткові напруги, що виникають в елементах колінчатих валів від крутильних, вигинистих і поздовжніх коливань, складають значну частину загальної напруги, яка знайдена без урахування коливань.

Зазвичай при розрахунку колінчатого вала на міцність за розрахунковий приймається номінальний режим роботи двигуна або режим максимального обертового моменту. Додаткове навантаження колінчатого вала від інерційних навантажень, пов'язаних з коливаннями, враховують за допомогою коефіцієнта динамічного посилення. При цьому вважається, що робота колінчатого вала відбувається поза зонами резонансних коливань, коли додаткові напруги досягають максимальних значень.

Протягом робочого циклу двигуна система навантажуючих колінчатий вал сил і моментів безперервно змінюється по куту повороту вала. Для оцінки міцності колінчатого вала слід розраховувати всю сукупність його положень за робочий цикл при відповідних величинах і напрямках діючих на вал зусиль.

Традиційно існуючі методи розрахунку колінчатого вала діляться на дві групи. В одній із них колінчатий вал багатоциліндрового двигуна розглядається як розрізна

система. Вал представляється набором колін, навантажених відповідними зусиллями. Вплив сусідніх колін враховується тільки через набігаючий обертальний момент. Даний метод досить простий, тому що розрахункова схема статично визначена.

Однак розрізна схема не враховує дію навантажень на коліна, які суміжні з даним і вплив надопорних згинальних моментів, які діють у різних площинах, що в ряді випадків істотно завищує рівень одержуваних при розрахунку напружень в порівнянні з експериментальними даними, отриманими в результаті тензометрування (тензометрія – сукупність експериментальних методів визначення механічної напруги деталі, конструкції, заснована на визначенні деформацій або інших параметрів матеріалу, викликаних механічною напругою) валів на працюючих двигунах.

Метод розрахунку вала як нерозрізної системи, коли розглядається цілком весь вал з прикладеними до всіх колін навантаженнями і при різних положеннях вала, є більш складним і в той же час більш трудомістким. Для реалізації закладених у методі можливостей у належній мірі, при розрахунку необхідно враховувати податливість опор корінних підшипників колінчатого вала, яка в свою чергу пов'язана з деформацією картера, підвісок і вкладишів підшипників, гідродинаміки масляного шару та ін.. Крім того, згинальна жорсткість колінчатих валів різна в залежності від кута нахилу площини, що проходить через вісь обертання, в якій розглядається вигин. Облік перерахованих факторів вимагає попереднього аналізу конструкції не тільки вала, а й пов'язаних деталей і вузлів. Тому часто розрахунок доцільно починати за допомогою розрізної схеми, особливо при попередньому аналізі різних варіантів конструкції, із переходом до нерозрізної схеми на наступному етапі розробки колінчатого вала.

1.1 Дослідження актуальності проблеми

Двигуни внутрішнього згоряння – найбільш поширений на сьогоднішній день тип двигуна, що встановлюється на різного роду техніку. Найпоширенішим видом ДВЗ є поршневий двигун, що широко використовується в силових агрегатах автомобілів, кораблів та ін.. Найбільш відповідальним, складним і навантаженим

елементом поршневого двигуна є колінчатий або кривошипний вал. Незважаючи на те, що принципові питання оцінки міцності колінчатих і кривошипних валів розроблені досить повно, методики розрахунку, які застосовуються в даний час є застарілими та не володіють необхідною для практичних завдань точністю. Це пояснюється тим, що зважаючи на складність форми колінчатих та кривошипних валів та одночасного впливу на них декількох змінних за величиною і напрямком сил, потрібно аналізувати розподіл напружень у великому числі розрахункових перетинів при різних кутах повороту вала. Наприклад, у порівняно простому двохколінному двохопорному валі необхідно провести аналіз розподілу напружень в 10-12 розрахункових перетинах на вплив від 6-8 змінних навантажень, які залежать від кута повороту вала та змінюються. Такі умови вимагають виконання обчислень в обсягах недоступних при «ручних» розрахунках. Проте, велика частина автомобільних двигунів були розроблені та виготовлені 30-40 років тому, коли «ручні» розрахунки були практично єдиним засобом аналізу. Розрахунки виконувалися за спрощеними методиками з численними наближеннями «в запас», що часто призводило до невиправдано високих запасів міцності та збільшення матеріаломісткості конструкцій. Уточнення та деталізація оцінки міцності та залишкового ресурсу колінчатих валів діючих двигунів може бути обґрунтуванням збільшення термінів їх експлуатації.

Великі можливості для уточнення й деталізації розрахунків на міцність мають сучасні інформаційні технології. Поява загально-математичних (MathCAD, MAPLE) і спеціалізованих (ALGOR, NASTRAN, ANSYS) програмних обчислювальних комплексів дозволяють визначати експертну оцінку міцності технічних об'єктів практично будь-якого ступеня складності. Наприклад, використання спеціалізованих програмних комплексів (ПК) для визначення напружено-деформованого стану кривошипів великих компресорів по методу кінцевих елементів (чисельний метод рішення диференціальних рівнянь із частковими похідними, а також інтегральних рівнянь, що виникають при вирішенні завдань прикладної фізики), дало можливість враховувати вплив дефектів і пошкоджень, виявлених в щоках, на міцність і довговічність корінного вала.

Практична значимість проекту полягає в можливості точно розраховувати мінімальні запаси втомної міцності колінчатих валів при проектуванні за мінімальних часових витрат. Застосування методу дозволяє скоротити витрати на виготовлення дослідних зразків колінчатих валів і проведення реальних експериментів по їх доведенню.

Отже, для підвищення ефективності розрахунків і швидкості проектування було вирішено створити інформаційне та програмне забезпечення системи розрахунку колінчатого вала.

1.2 Аналіз аналогів

На сьогоднішній день існує декілька програмних комплексів, за допомогою яких можна виконати розрахунок колінчатого вала, але всі вони занадто громіздкі, чи є складними у використанні, або не є безкоштовними. Найбільш схожими з розроблюваним проектом ПК є MSC Nastran, ANSYS і KVAL.

1.2.1 MSC Nastran

MSC Nastran – це сучасна розрахункова система з тісною інтеграцією з іншими системами високого рівня (рис. 1.4). Вона забезпечує повний набір розрахунків, включаючи розрахунок напружено-деформованого стану, запасів міцності, власних частот і форм коливань, аналіз стійкості, дослідження сталих і несталих динамічних процесів, рішення задач теплопередачі, акустичних явищ, нелінійних статичних і нелінійних перехідних процесів, аналіз складної контактної взаємодії, розрахунок критичних частот і вібрацій роторних машин, аналіз частотних характеристик при впливі випадкових навантажень і імпульсного широкосмугового впливу, дослідження аеропружності на дозвукових і надзвукових швидкостях.

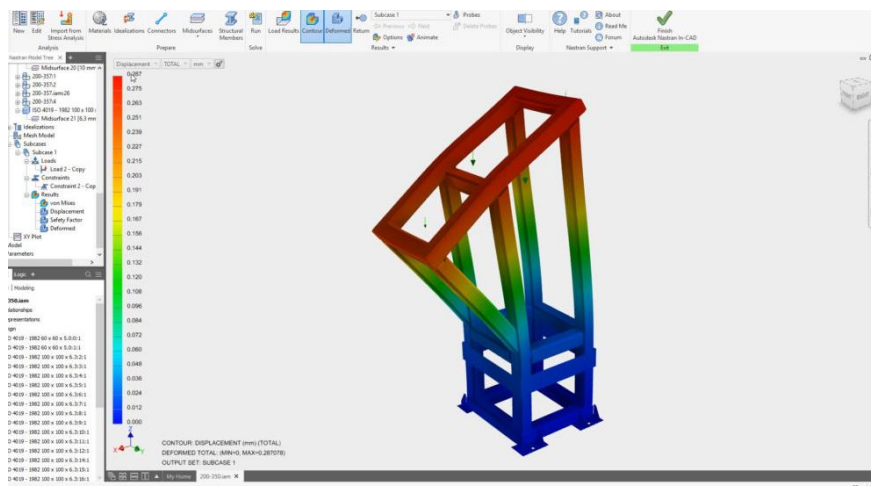


Рисунок 1.4 – Робоче вікно програми «MSC Nastran»

Передбачена можливість моделювання практично всіх типів матеріалів, включаючи композитні та гіперпружні. До складу розширених функцій входить технологія суперелементів (підконструкцій), включаючи просунуті методи динамічних конденсацій, модальний синтез і розвинені методи аналізу динаміки складних структур на основі суперелементів і формулювань методу Крейга-Бемптона.

MSC Nastran включає спеціальні елементи для моделювання точкового та шовного зварювання, болтових і заклепкових з'єднань (з'єднання деталей за допомогою заклепок), набір жорстких елементів різних типів і т. д. Макромова DMAP дозволяє користувачам розробляти власні спеціальні модулі, програми та включати їх в MSC Nastran, створюючи абсолютно нові рішення на потужній математичній базі MSC Nastran.

MSC Nastran працює на персональних комп'ютерах, робочих станціях, кластерах, суперкомп'ютерах та передбачає можливості векторної та паралельної обробки даних на обчислювальних комплексах, які підтримують дані функції.

Головним недоліком даного ПК є потреба в придбанні ліцензії.

1.2.2 ANSYS Mechanical Enterprise

Програмні продукти ANSYS для механіки деформованого твердого тіла дозволяють вирішувати широкий спектр інженерних задач в області міцності та теплових розрахунків, а також приймати конструкторські рішення швидше й

ефективніше. За допомогою інструментів ANSYS користувачі мають можливість не тільки автоматизувати й адаптувати власні розрахунки, але й проводити параметризовані дослідження безлічі варіантів конструкції для пошуку найкращого рішення. Програмні продукти ANSYS для розрахунків на міцність можуть бути легко пов'язані з іншими розрахунковими пакетами для проведення міждисциплінарних розрахунків і аналізу поведінки складних конструкцій в умовах, які є максимально близькими до реальних.

ANSYS Mechanical Enterprise є найбільш технологічною ліцензією для проведення розрахунків на міцність і максимально відповідає високим вимогам розрахункових відділів промислових підприємств (рис. 1.5).

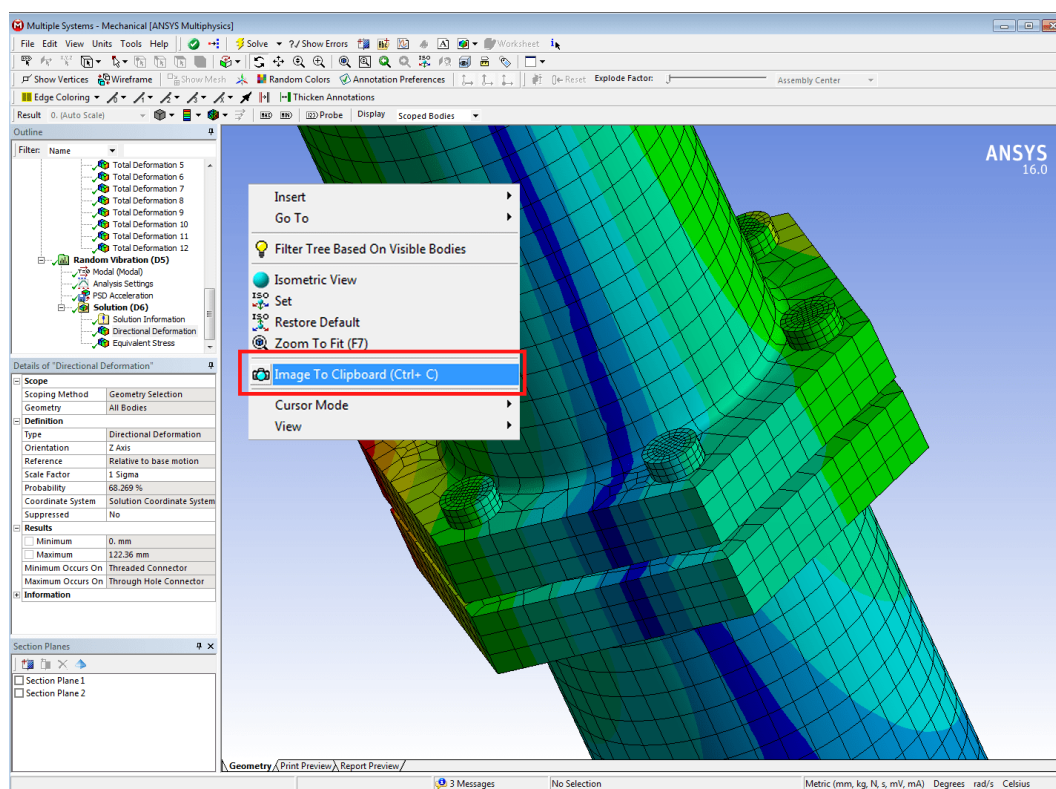


Рисунок 1.5 – Робоче вікно програми «ANSYS Mechanical Enterprise»

Інструменти ANSYS Mechanical Enterprise дозволяють аналізувати всі види напружено-деформованого стану, проводити теплові розрахунки та розрахунки деталей на зношення, моделювати різноманітну нелінійну поведінку матеріалів та досліджувати динамічні характеристики конструкцій у зручному та високопродуктивному розрахунковому середовищі.

ANSYS Mechanical Enterprise дозволяє проводити всі види розрахунків на міцність, включаючи статичний, на зношення, модальний, гармонійний, спектральний, випадкові вібрації і аналіз перехідних процесів, а також моделювати лінійну та нелінійну втрату стійкості несучих елементів і вузлів (рис. 1.6). Для поглибленого аналізу поведінки конструкції доступні спеціалізовані інструменти для моделювання, розрахунку роторної динаміки, метод суперелементів, моделювання тріщин у явному вигляді, циклічна симетрія, адаптивне перебудування сітки і т.д.

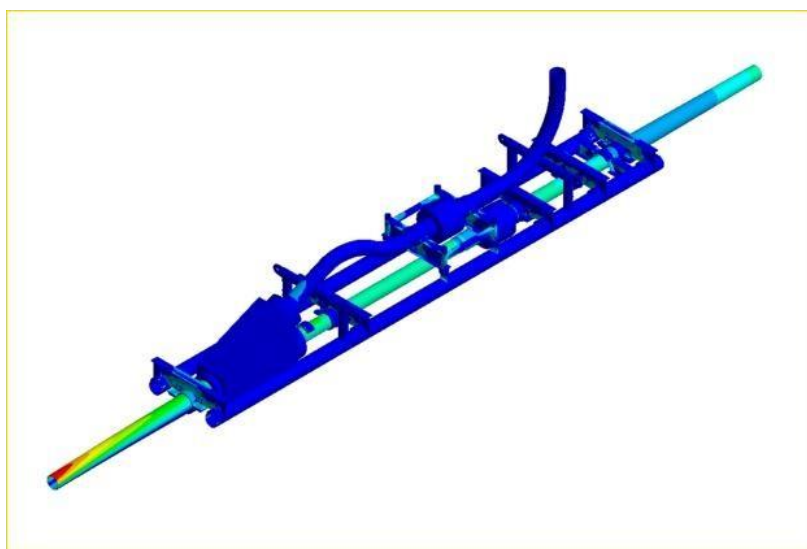


Рисунок 1.6 – Робоче вікно розрахунку механічних напружень в опорі трубопроводу в програмі «ANSYS Mechanical Enterprise»

Програмні продукти, які входять до складу ANSYS Mechanical Enterprise, інтегровані в розрахункову платформу ANSYS Workbench, що служить для проведення параметричних і міждисциплінарних розрахунків. Для досягнення максимальної ефективності при проведенні розрахунків на міцність ANSYS Mechanical Enterprise також включає в себе програмні продукти ANSYS DesignXplorer для проведення багатокритеріальної оптимізації, ANSYS SpaceClaim для підготовки геометричних моделей, ANSYS AQWA для розрахунку впливу хвиль і ANSYS ACT для адаптації функціоналу та розробки спеціалізованих додатків.

Головними недоліками даного ПК є потреба в придбанні ліцензії, а також його громіздкість і вибагливість до апаратного устаткування.

1.2.3 KVAL

Програмний комплекс KVAL призначений спеціально для розрахунку колінчатого вала (рис. 1.7).

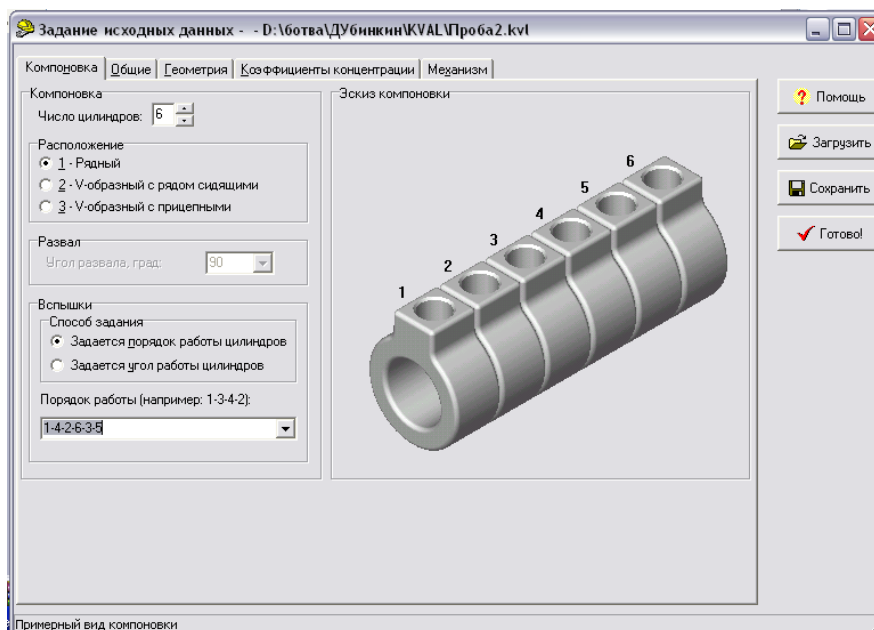


Рисунок 1.7 – Рабочее вікно програми «KVAL»

Дана програма є найближчим аналогом продукту, що розробляється. Головним мінусом даного ПК є його відсутність у відкритому доступі – він використовується лише студентами Московського державного технічного університету імені М. Е. Баумана.

1.3 Постановка задачі

1.3.1 Мета задачі дослідження

Тема дипломного проекту – «Інформаційне та програмне забезпечення системи розрахунку колінчатого вала».

Метою дипломного проекту є створення програмного додатку для розрахунку колінчатого вала (корінної шийки, шатунної шийки та щік), що підвищить ефективність розрахунків та прискорить проектування колінчатих валів.

Задачами дипломного проекту є наступні:

- пошук, аналіз та структуризація інформації по розрахунку колінчатих валів;
- дослідження галузі машинобудування;
- ознайомлення з аналогами створюваного проекту та їх аналіз;
- вибір засобів для програмної реалізації;
- реалізація програмного додатку для розрахунку колінчатого вала;
- тестування роботи розробленого продукту та аналіз результатів.

2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

В якості платформи для розробки програми була обрана операційна платформа WIN32, яка зустрічається практично на всіх сучасних настільних системах. Цей вибір обумовлений тим, що операційні системи Microsoft – сімейство найбільш популярних систем, що володіє простим зовнішнім оформленням, яке спрощує взаємодію з кінцевим користувачем.

На сьогоднішній день на програмну індустрію впливають такі мови програмування: C++, C#, Python, JavaScript, Visual Basic. Кожну з них можна використовувати як алгоритмічну.

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (при компанії Microsoft).

Синтаксис C# близький до C++ і Java. мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато що від попередників – мов C++, Delphi, Modula і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, множинне спадкування класів (на відміну від C++).

Серед переваг мови C# можна виділити такі:

- C# добре розвивається завдяки зусиллям корпорації Microsoft;
- C# має безкоштовні для приватних осіб і невеликих компаній інструменти (Visual Studio, Xamarin);
- містить багато «синтаксичного цукру». «Синтаксичний цукор» – це такі конструкції, які створені для полегшення написання та розуміння коду (особливо, якщо це код іншого розробника), вони ігноруються при компіляції;
- має середній поріг входження. Синтаксис схожий на C, C++ або Java і полегшує перехід для розробників з інших мов програмування;
- додано функціональне програмування (F #);
- велика спільнота розробників.

Проте C# також має наступні недоліки:

- орієнтованість, загалом, тільки на .NET (Windows платформа);
- безкоштовність тільки для невеликих компаній, учнів і розробників-одинаків. Для великих команд придбання ліцензій є досить дорогим;
- використовує застарілий оператор «go to».

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і кращого читання коду. Синтаксис ядра Python – мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване програмування. Основні архітектурні риси – це динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція (можливість визначити тип і структуру об'єкта під час виконання програми), механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, у свою чергу, можуть об'єднуватися в пакети.

Python – мова програмування, яка активно розвивається, нові версії з додаванням чи зміною мовних властивостей виходять приблизно раз в два з половиною роки.

JavaScript – мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі.

Мова зазвичай використовується як вбудована для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерях, як мова сценаріїв для додання інтерактивності веб-сторінкам, також використовується для написання прикладного програмного забезпечення.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістів. Мовою

JavaScript не володіє будь-яка компанія або організація, що відрізняє його від ряду мов програмування, використовуваних у веб-розробці.

Visual Basic – це мова програмування, яка розробляється корпорацією Microsoft. Мова частково успадкувала синтаксис попередника – мови BASIC, у якого є чимало діалектів. У той же час Visual Basic поєднує в собі процедури та елементи об'єктно-орієнтованих і компонентно-орієнтованих мов програмування.

Visual Basic також призначений для швидкої розробки (RAD) додатків баз даних для операційних систем сімейства Microsoft Windows. Велика кількість готових компонентів надає можливість розробнику відразу ж почати розробляти бізнес-логіку додатку, не відволікаючи його увагу на написання коду запуску програми, підписки на події та інших механізмів, які Visual Basic реалізує автоматично.

C++ – компільована, статично типізована мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова має багату стандартну бібліотеку, яка включає в себе поширені контейнери та алгоритми, введення-виведення, регулярні вирази, підтримку багатопоточності та інші можливості. C++ поєднує властивості як високорівневих, так і низькорівневих мов, на відміну від попередника – мови C, де найбільшу увагу приділено було підтримці об'єктно-орієнтованого й узагальненого програмування.

C++ Builder – програмний продукт, інтегроване середовище програмування (IDE), інструмент швидкої розробки додатків (RAD), система, яка використовується розробниками для створення програмного забезпечення на мові програмування C++.

C++ Builder об'єднує в собі комплекс об'єктних бібліотек (STL, VCL, CLX, MFC і ін.), компілятор, відладчик, редактор коду та інші компоненти.

C++ Builder містить інструменти, які за допомогою drag-and-drop дійсно роблять розробку візуальною, спрощує програмування завдяки вбудованому WYSIWYG – редактору інтерфейсу та ін..

Після аналізу існуючих технологій, було обрано мову C++ та інтегроване середовище розробки C++ Builder, тому що відсутня потреба у вивченні «з нуля» даної мови та середовища програмування, що значно прискорить розробку

програмного продукту. Важливими критеріями стали також властивості мови, а саме: велика безпечність, можливість створення узагальненого коду за допомогою шаблонів, можливість використовувати об'єктно-орієнтований підхід, спрощення коду за рахунок перевантаження функцій і операторів, більш проста обробка помилок за рахунок винятків, масштабованість. Середовище програмування було обрано через високу продуктивність, інкрементну компіляцію та компоновку виконуваного модуля, невибагливість до апаратних ресурсів.

3 ПРАКТИЧНИЙ РОЗДІЛ

3.1 Визначення структури додатку

Структура розроблюваного додатку включатиме наступні елементи: сам додаток, файли .txt та .tsv для збереження результатів та всі інші файли, вміст яких можна переглянути. Блок-схема додатку представлена у додатку А.

Структура додатку зображена на рисунку 3.1 :



Рисунок 3.1 – Структура додатку

Розробка структури додатку також включає створення шаблону, який було розроблено відповідно до поставленої задачі. Він включає заголовочний блок, блок меню, у якому розміщуються кнопки вибору можливих дій із формою. Нижче розташовані вкладки програми. Далі розташований блок у якому відбувається ввід даних та відображуються результати пошуку (рис. 3.2).

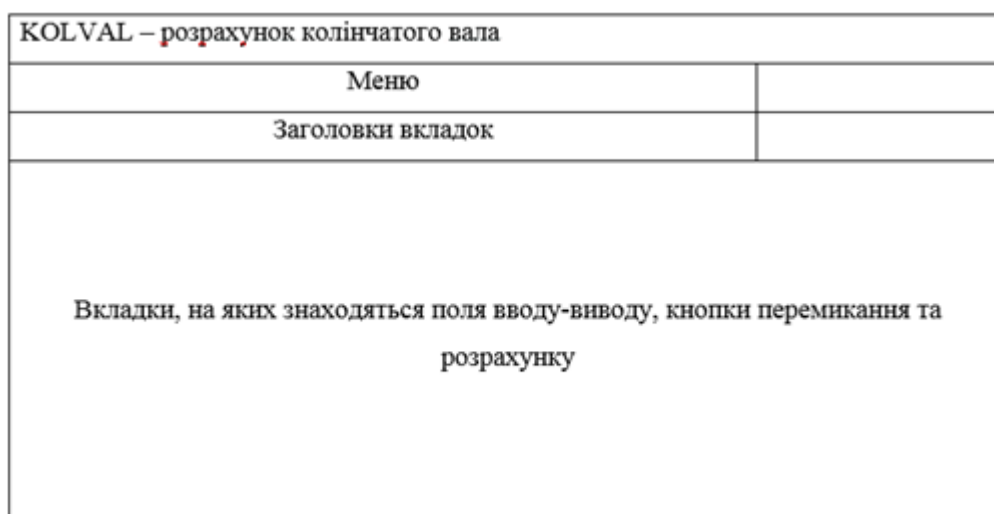


Рисунок 3.2 – Шаблон додатку

3.2 Розробка дизайну додатку

Наступним кроком була розробка інтерфейсу користувача та дизайну. Інтерфейс програмного додатку представлено на рисунках 3.3-3.7:

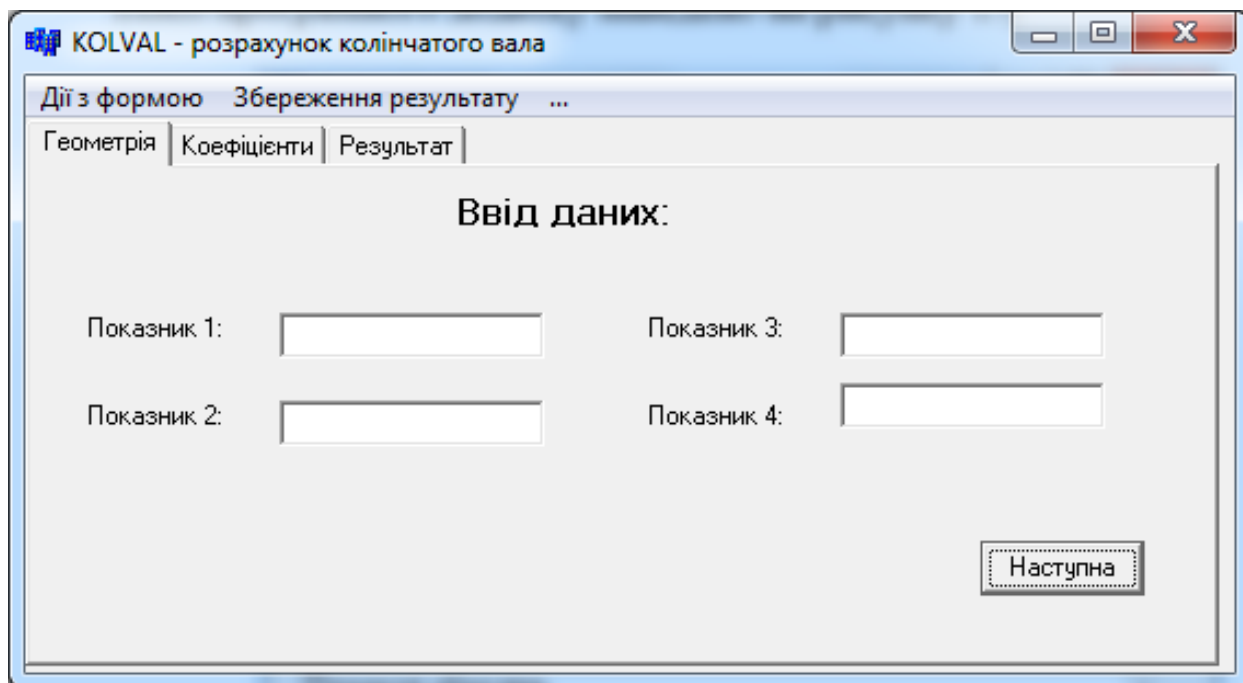


Рисунок 3.3 – Інтерфейс додатку (частина 1)

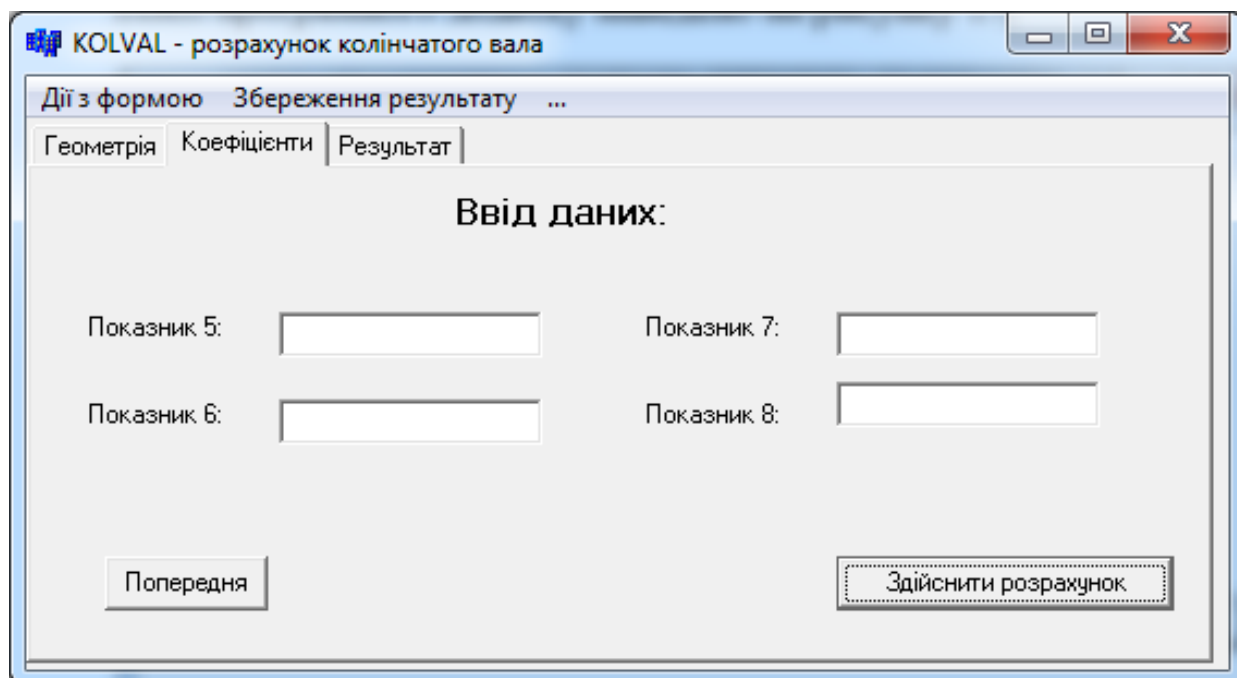


Рисунок 3.4 – Інтерфейс додатку (частина 2)

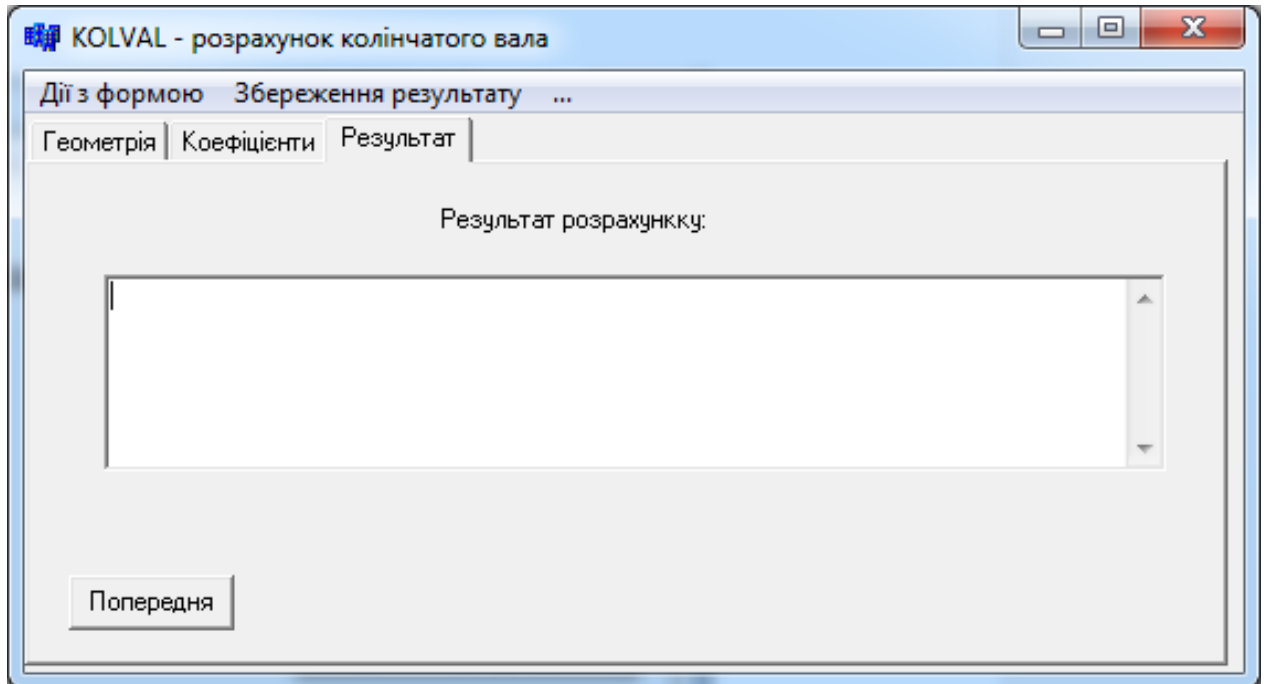


Рисунок 3.5 – Інтерфейс додатку (частина 3)

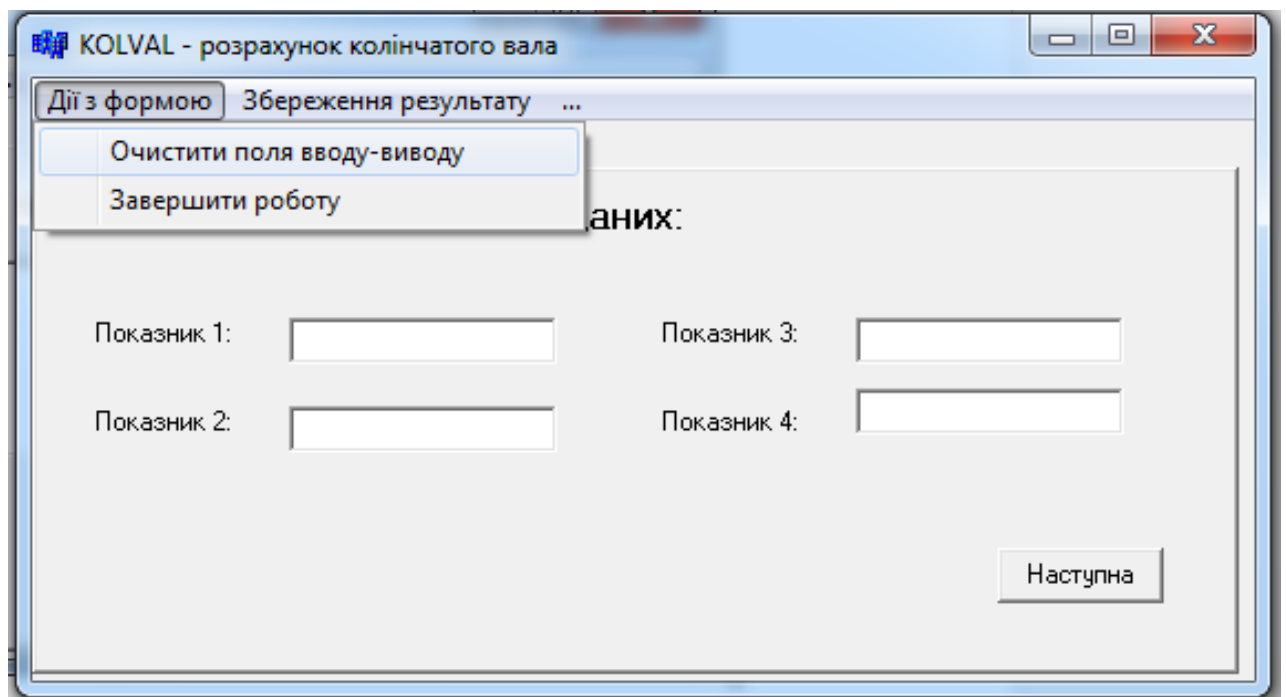


Рисунок 3.6 – Інтерфейс додатку (частина 4)

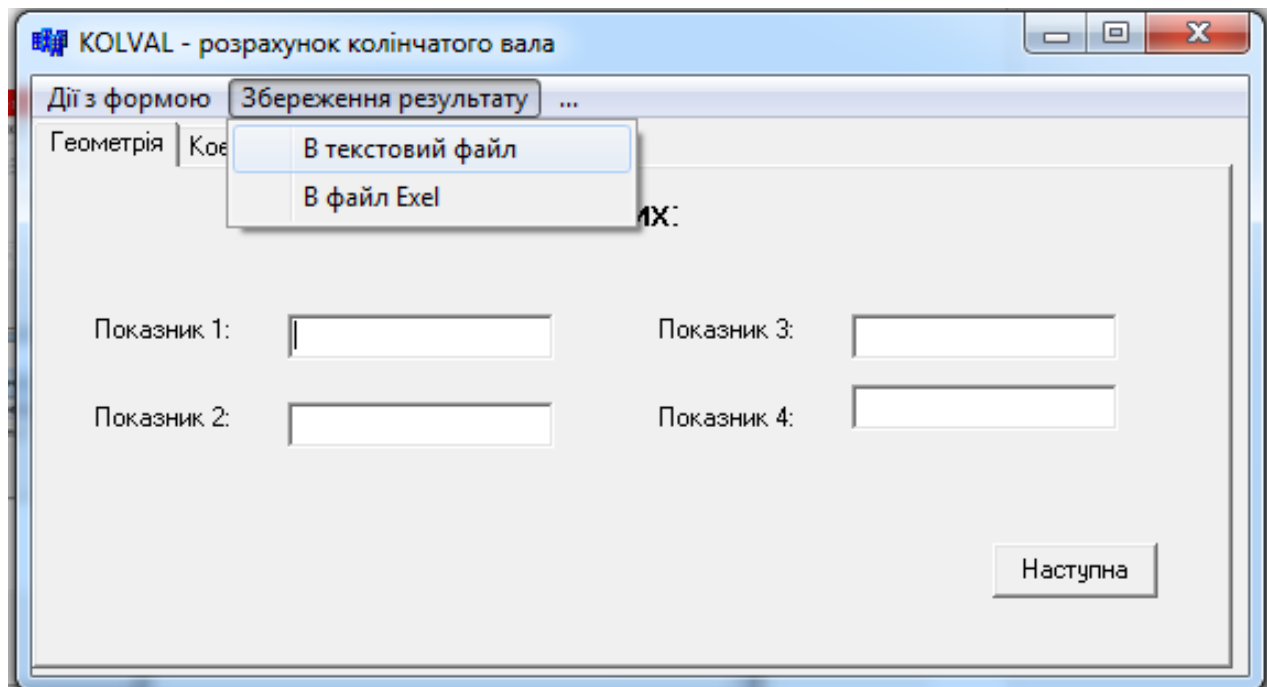


Рисунок 3.7 – Інтерфейс додатку (частина 5)

3.3 Розробка функціоналу додатку

Далі буде розглянуто роботу з додатком. Повний лістинг коду представлений у Додатку Б.

Виконання розрахунку (якщо всі поля вводу заповнені) представлено на рисунку 3.8.

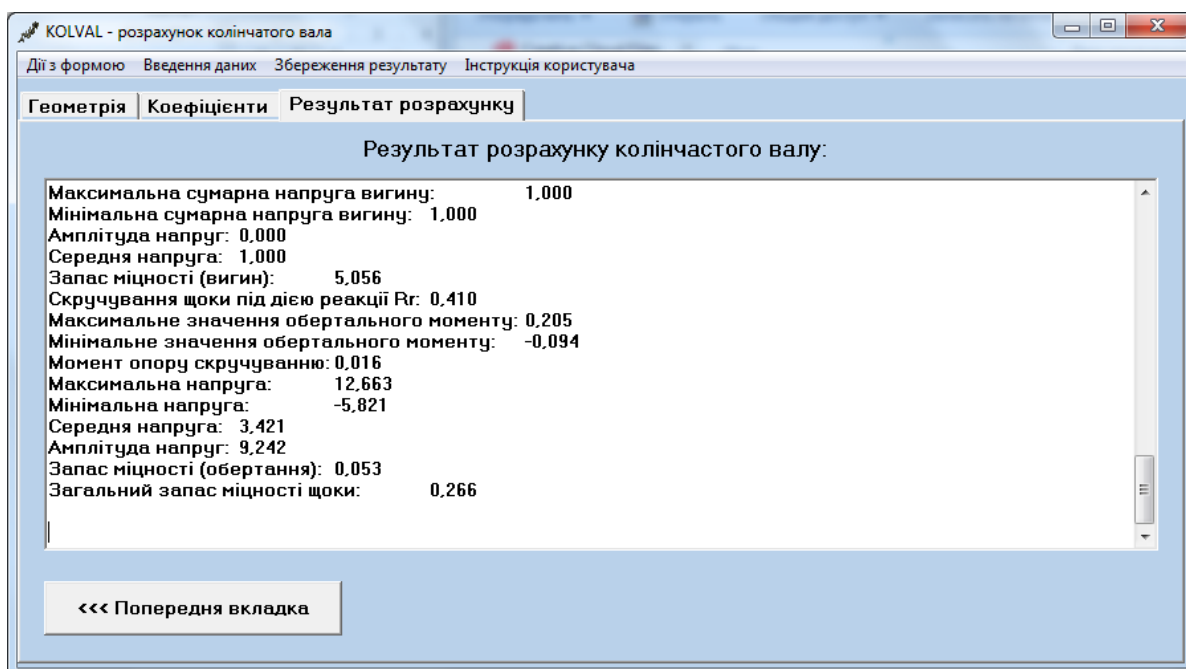


Рисунок 3.8 – Результат роботи програми

Збереження результату в текстовий файл представлено на рисунку 3.9 та файл Excel (за умови, що поле виводу містить дані), представлений на рисунку 3.10.

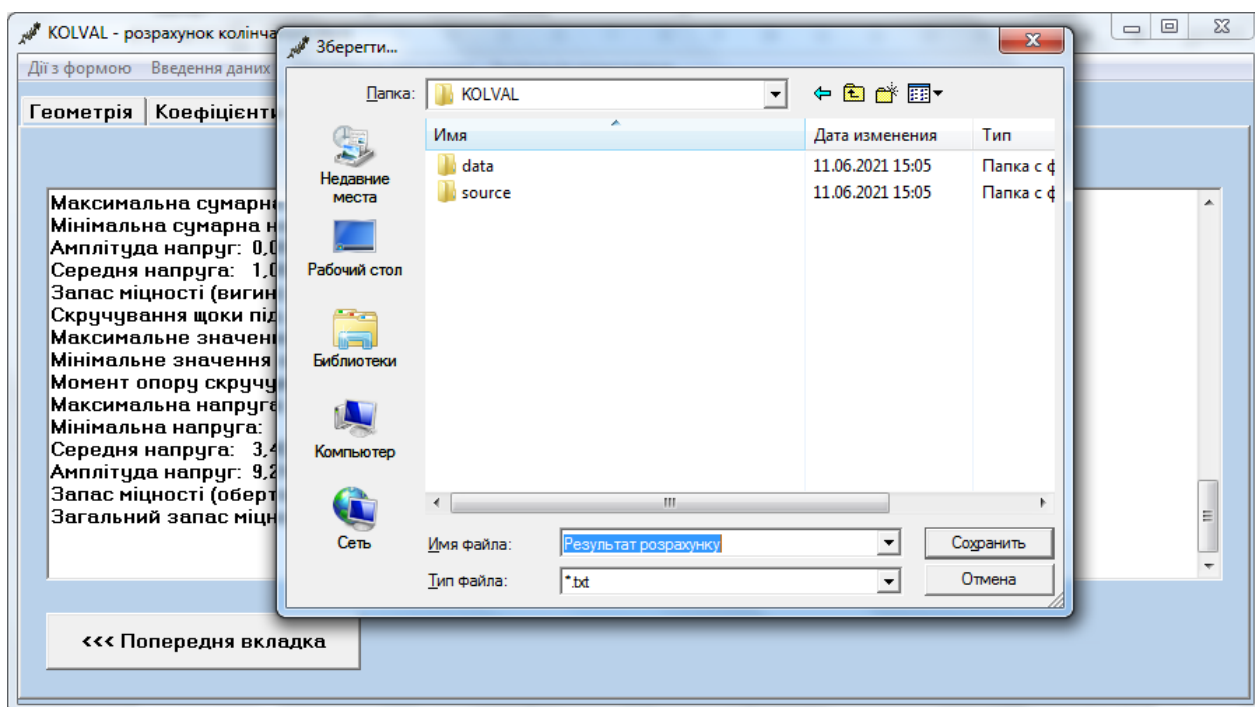


Рисунок 3.9 – Збереження результату в txt-файл

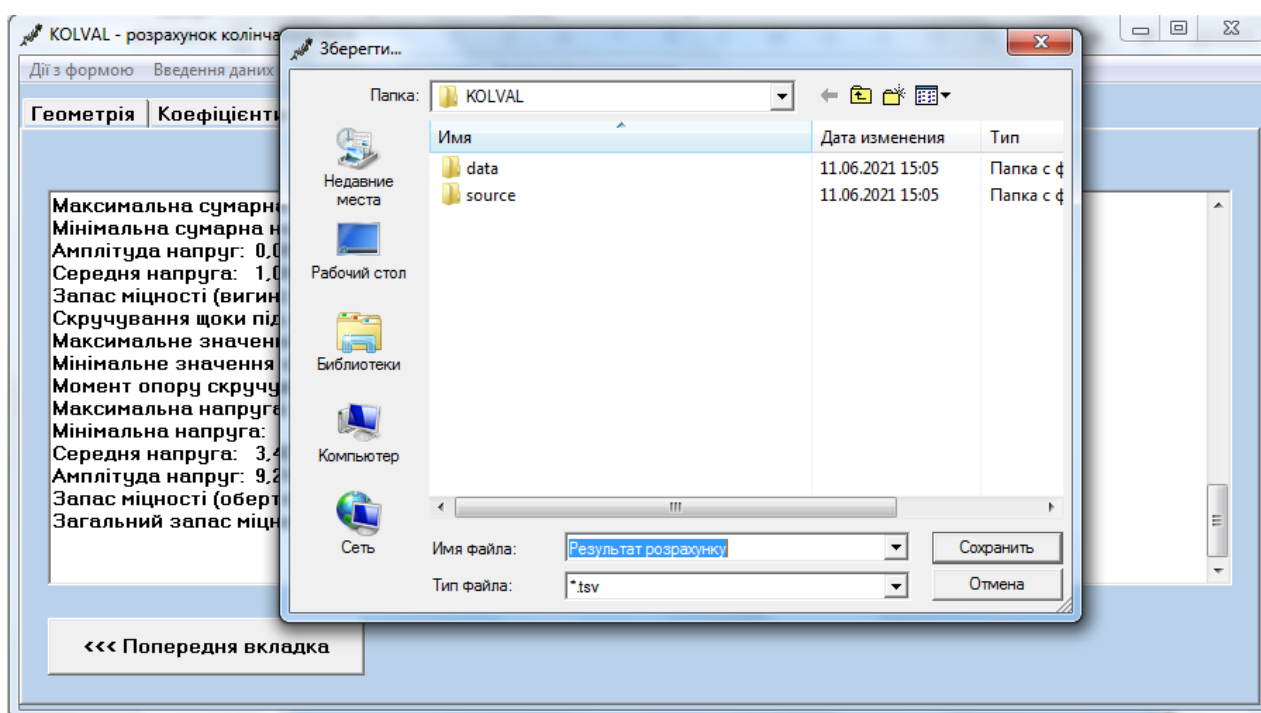


Рисунок 3.10 – Збереження результату в Excel-файл

Додаткова опція – перегляд вмісту файлу, зображена на рисунку 3.11. Таким чином можна відкрити для перегляду та редагування будь-який файл.

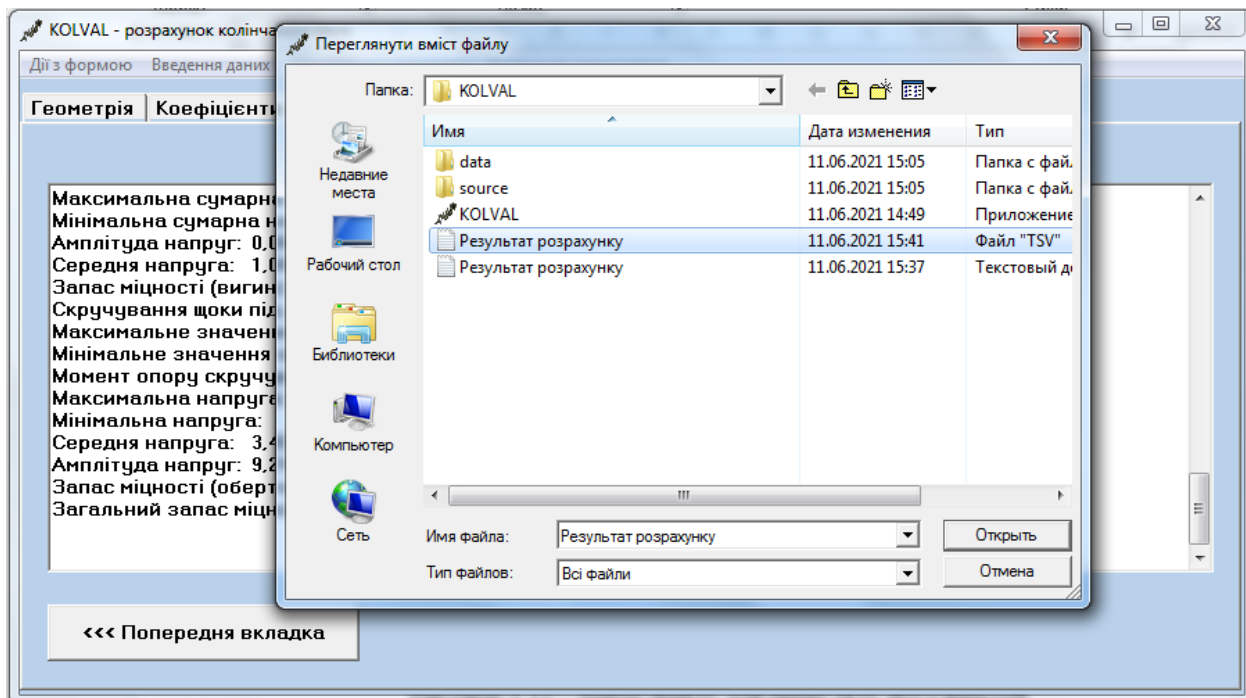


Рисунок 3.11 – Вибір файлу для перегляду його вмісту

Додаткові опції програми – очистити поля вводу-виводу та завершення роботи програми представлено на рисунку 3.12.

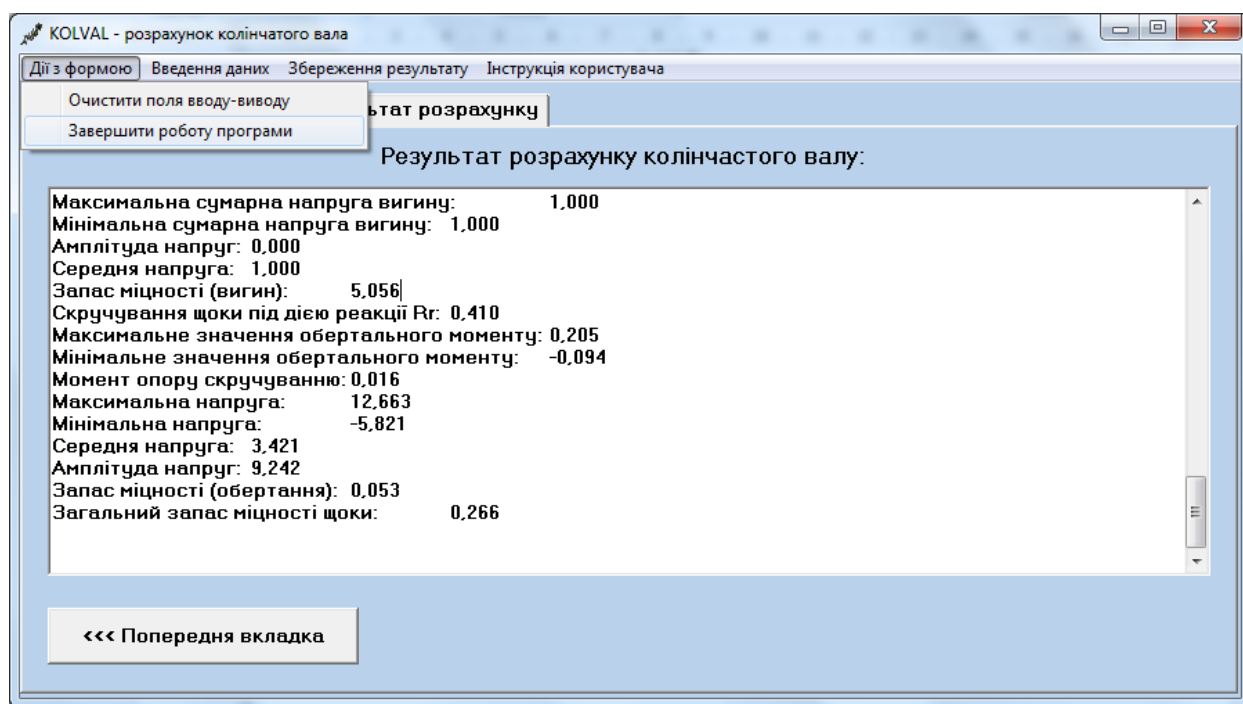


Рисунок 3.12 – Додаткові опції програми

Даний додаток передбачає введення початкових даних. Окрім ручного вводу передбачено додатково зчитування з текстового файлу, представлено на рисунку 3.13.

Рисунок 3.13 – Заповнення полів вводу при зчитуванні даних з файлу

Треба зазначити, що кількість числових даних може бути меншою за кількість полів вводу. У такому випадку буде заповнено стільки полів, на скільки вистачить величин, інші поля вводу залишаться порожніми. Також передбачене автозаповнення наявних полів вводу випадковими величинами, представлено на рисунку 3.14.

Рисунок 3.14 – Заповнення полів вводу при виборі автозаповнення

3.4 Тестування програмного додатку

Після створення прикладного програмного додатку «Розрахунок колінчатого вала» його було протестовано.

Якщо поле вводу не має значення, то при спробі виконати обчислення буде видано повідомлення, яке представлено на рисунку 3.15.

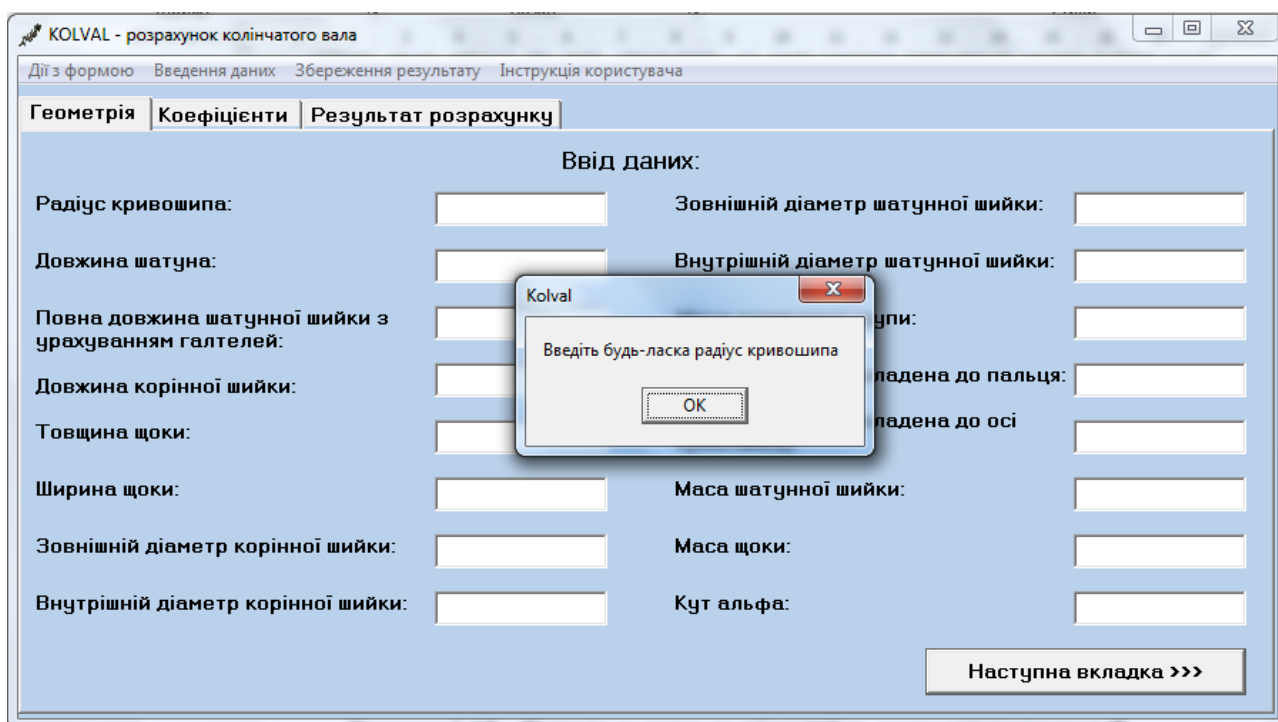


Рисунок 3.15 – Реакція додатку на відсутність значення поля вводу

Після закриття повідомлення буде встановлено курсор в відповідне поле вводу.

Якщо поле виводу порожнє, то при спробі зберегти результат буде видано повідомлення, зображене на рисунку 3.16.

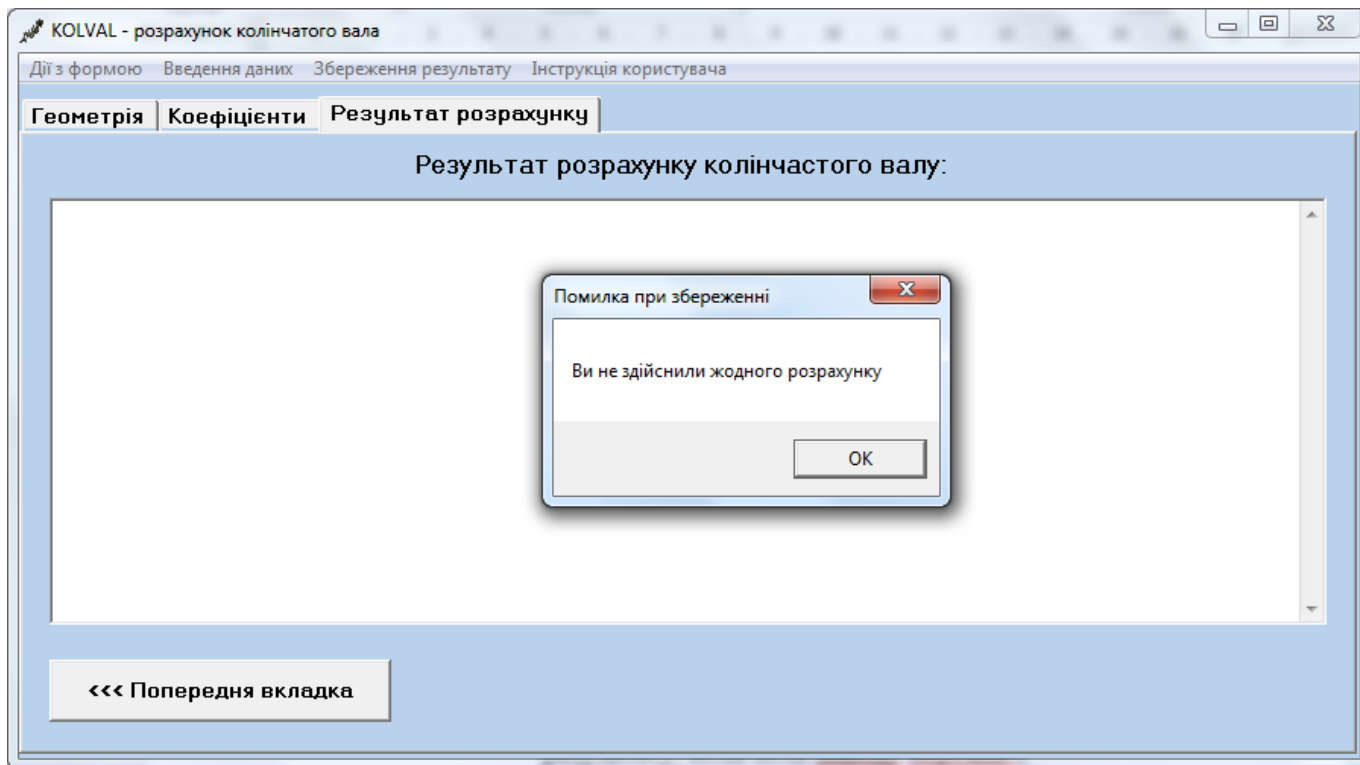


Рисунок 3.16 – Реакція додатку при збереженні результату, коли поле виводу порожнє

За допомогою події OnKeyPress було заборонено ввід всіх символів крім цифр та коми. Якщо спробувати ввести замість цифр інші символи – поле вводу залишиться пустим, або не зміниться (якщо дані вже були введені).

Після тестування розробленого додатку помилок у роботі не було виявлено.

ВИСНОВОК

У результаті виконання дипломного проекту була досліджена галузь машинобудування, велика увага приділялась питанню розрахунку колінчатого вала. Також були проаналізовані існуючі розрахункові програми, визначено їх функції, якими вони володіють, а також переваги та недоліки.

Було розглянуто аналіз сучасних засобів реалізації та обрано мову програмування C++ та середовище розробки C++ Builder, тому що відсутня потреба у вивченні даної мови та середовища програмування спочатку, що значно прискорило розробку програмного продукту.

Було проаналізовано сучасні вимоги та способи розрахунку колінчатих валів. Згідно з отриманою інформацією для програмної реалізації найкраще підходить метод розрахунку вала як розрізної системи.

Одним із завдань дипломного проекту було дослідження процесу автоматизації розрахунку колінчатого вала. Вирішенням даної проблеми стало розробка програмного додатку з можливістю розрахунку корінної шийки, шатунної шийки та щік колінчатого вала. Це дозволило підвищити ефективність та швидкість проектування колінчатих валів та ДВЗ загалом. Завдяки реалізації поставленого завдання інженери мають доступ до вузько-направленої програми по розрахунку колінчатих валів.

У результаті дипломного проектування було розроблено та протестовано розрахунковий додаток. Усі тест-кейси були пройдені успішно. Серед них можна виокремити перевірку полів на наявність значень та перевірку поля виводу на наявність результату перед збереженням у зовнішній файл.

СПИСОК ЛІТЕРАТУРИ

1 Боровських Ю. І., Буральов Ю. В., Морозов К. А. Будова автомобілів: навчальний посібник / Ю. І. Боровських, Ю. В. Буральов, К. А. Морозов. – К.: Вища школа, 1991. – 304 с.

2 Двигатели внутреннего сгорания: В 3 кн. Кн. 2. Динамика и конструирование двигателей: Учебник для вузов/ В.Н. Луканин, М.Г. Шатрова, и др.; Под ред. В.Н. Луканина, М.Г. Шатрова.– М.: Высш. школа, 2007.

3 Динамика поршневых двигателей / Под ред. А.И. Ямаина, А.В. Жарова. – М., 2003. 4. Автомобильные и тракторные двигатели: Учебное пособие к курсовому проектированию / Под ред. В.Н. Уханова. – Пенза, 2005.

4 Кисликов В. Ф., Луцик В. В. Будова й експлуатація автомобілів: Підручник. – 6-те вид. – К.: Либідь, 2006. – 400 с.

5 Расчет автомобильных и тракторных двигателей / Под ред. А.И. Колчанина, В.П. Демидова. – М.: Высш. школа, 2002.

6 Расчет коленчатого вала в программе KVAL. [Электронный ресурс]. – 2013. – Режим доступа до ресурсу: <http://kurs.znate.ru/docs/index-133260.html?page=8>.

7 Сирота В. І. Основи конструкції автомобілів. Навчальний посібник для вузів. К.: Арістей, 2005. – 280 с.

8 Шевченко П. Л. Расчет коленчатого вала [Электронный ресурс] / П. Л. Шевченко // СибАДИ. – 2008. – Режим доступа до ресурсу: <http://bek.sibadi.org/fulltext/ED1711.pdf>.

9 Архангельский А.Я. Язык С++ в С++ Builder. Справочное и методическое пособие / Архангельский А.Я., 2008. – 944 с. – (Бином).

10 Стивен П. Язык программирования С++. Лекции и упражнения 6-е изд. / Прата Стивен., 2015. – 1248 с. – (Диалектика).

ДОДАТОК А

Блок-схема додатку



Рисунок А.1 – Блок-схема розрахункового додатку

ДОДАТОК Б

Програмний код додатку

```
//-----
#include <vcl.h>
#pragma hdrstop
#include "math.h"
#include <Math.hpp>
#include <stdio.h>
#include "KOLVALUnit1.h"
#include <shellapi.h>
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::N4Click(TObject *Sender)
{
    Edit1->Clear();
    Edit2->Clear();
    Edit3->Clear();
    Edit4->Clear();
    Edit5->Clear();
    Edit6->Clear();
    Edit7->Clear();
    Edit8->Clear();
    Edit9->Clear();
    Edit10->Clear();
    Edit11->Clear();
    Edit12->Clear();
    Edit13->Clear();
    Edit14->Clear();
    Edit15->Clear();
    Edit16->Clear();
    Edit17->Clear();
    Edit18->Clear();
    Edit19->Clear();
    Edit20->Clear();
    Edit21->Clear();
    Edit22->Clear();
    Edit23->Clear();
    Edit24->Clear();
    Edit25->Clear();
    Edit26->Clear();
}
```

```

        Edit27->Clear();
        Edit28->Clear();
        Edit29->Clear();
        Edit30->Clear();
        Edit31->Clear();
        Mem01->Clear();
    }
//-----
void __fastcall TForm1::N5Click(TObject *Sender)
{
    Close();
}
//-----
void __fastcall TForm1::N2Click(TObject *Sender)
{
    if(Mem01->Lines->Count == 0)
    {
        Application->MessageBox("Ви не здійснили жодного
розрахунку", "Помилка при збереженні", MB_OK);
        return;
    }

    SaveDialog1->Filter = "*.txt|*.txt";
    SaveDialog1->Title = "Зберегти...";
    if(SaveDialog1->Execute())
    {
        Mem01->Lines->SaveToFile(SaveDialog1->FileName +
".txt");
    }
}
//-----
void __fastcall TForm1::Excel1Click(TObject *Sender)
{
    if(Mem01->Lines->Count == 0)
    {
        Application->MessageBox("Ви не здійснили жодного
розрахунку", "Помилка при збереженні", MB_OK);
        return;
    }

    SaveDialog1->Filter = "*.tsv|*.tsv";
    SaveDialog1->Title = "Зберегти...";
    if(SaveDialog1->Execute())
    {
        Mem01->Lines->SaveToFile(SaveDialog1->FileName +
".tsv");
    }
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet1;
    Mem01->Clear();
}

```

```

Edit1->Text = "0,31";
Edit2->Text = "0,83";
Edit3->Text = "0,91";
Edit4->Text = "0,57";
Edit5->Text = "0,18";
Edit6->Text = "0,96";
Edit7->Text = "30";
Edit8->Text = "0,3";
Edit9->Text = "0,33";
Edit10->Text = "1,2";
Edit11->Text = "45";
Edit12->Text = "12";
Edit13->Text = "0,15";
Edit14->Text = "0,7";
Edit15->Text = "0,64";
Edit16->Text = "0,64";
Edit17->Text = "0,22";
Edit18->Text = "0,64";
Edit19->Text = "0,48";
Edit20->Text = "0,2";
Edit21->Text = "0,42";
Edit22->Text = "0,91";
Edit23->Text = "0,86";
Edit24->Text = "0,15";
Edit25->Text = "0,1";
Edit26->Text = "0,53";
Edit27->Text = "0,72";
Edit28->Text = "0,8";
Edit29->Text = "0,17";
Edit30->Text = "0,82";
Edit31->Text = "0,52";
}
//-----

void __fastcall TForm1::N7Click(TObject *Sender)
{
    FILE *fp;
    fp = fopen("data/data.txt","r");
    if(fp == NULL)
    {
        ShowMessage("Файл data.txt відсутній в директорії
додатку");
        return;
    }

    char a[31][10] = {"", "", "", "", "", "", "", "", "", "", "",
"", "", "", ""};
    int i = 0;

    while(!feof(fp))
    {
        fscanf(fp, "%s", &a[i]);
        i++;
    }
}

```

```

}

Edit1->Text = a[0];
Edit2->Text = a[1];
Edit3->Text = a[2];
Edit4->Text = a[3];
Edit5->Text = a[4];
Edit6->Text = a[5];
Edit7->Text = a[6];
Edit8->Text = a[7];
Edit9->Text = a[8];
Edit10->Text = a[9];
Edit11->Text = a[10];
Edit12->Text = a[11];
Edit13->Text = a[12];
Edit14->Text = a[13];
Edit15->Text = a[14];
Edit16->Text = a[15];
Edit17->Text = a[16];
Edit18->Text = a[17];
Edit19->Text = a[18];
Edit20->Text = a[19];
Edit21->Text = a[20];
Edit22->Text = a[21];
Edit23->Text = a[22];
Edit24->Text = a[23];
Edit25->Text = a[24];
Edit26->Text = a[25];
Edit27->Text = a[26];
Edit28->Text = a[27];
Edit29->Text = a[28];
Edit30->Text = a[29];
Edit31->Text = a[30];
}
//-----
void __fastcall TForm1::N9Click(TObject *Sender)
{
    OpenFileDialog->Filter = "Всі файли|*.*";
    OpenFileDialog->Title = "Переглянути вміст файлу";
    if(OpenDialog1->Execute())
        ShellExecute(Handle, NULL, OpenFileDialog-
>FileName.c_str(), NULL, NULL, SW_NORMAL);
}
//-----
void __fastcall TForm1::N8Click(TObject *Sender)
{
    float a[31];

    for(int i = 0; i < 31; i++)
        a[i] = (rand()%100 + 1)/10.0;

    Edit1->Text = RoundTo(a[0], -2);
    Edit2->Text = RoundTo(a[1], -2);
}

```

```

Edit3->Text = RoundTo(a[2], -2);
Edit4->Text = RoundTo(a[3], -2);
Edit5->Text = RoundTo(a[4], -2);
Edit6->Text = RoundTo(a[5], -2);
Edit7->Text = RoundTo(a[6], -2);
Edit8->Text = RoundTo(a[7], -2);
Edit9->Text = RoundTo(a[8], -2);
Edit10->Text = RoundTo(a[9], -2);
Edit11->Text = RoundTo(a[10], -2);
Edit12->Text = RoundTo(a[11], -2);
Edit13->Text = RoundTo(a[12], -2);
Edit14->Text = RoundTo(a[13], -2);
Edit15->Text = RoundTo(a[14], -2);
Edit16->Text = RoundTo(a[15], -2);
Edit17->Text = RoundTo(a[16], -2);
Edit18->Text = RoundTo(a[17], -2);
Edit19->Text = RoundTo(a[18], -2);
Edit20->Text = RoundTo(a[19], -2);
Edit21->Text = RoundTo(a[20], -2);
Edit22->Text = RoundTo(a[21], -2);
Edit23->Text = RoundTo(a[22], -2);
Edit24->Text = RoundTo(a[23], -2);
Edit25->Text = RoundTo(a[24], -2);
Edit26->Text = RoundTo(a[25], -2);
Edit27->Text = RoundTo(a[26], -2);
Edit28->Text = RoundTo(a[27], -2);
Edit29->Text = RoundTo(a[28], -2);
Edit30->Text = RoundTo(a[29], -2);
Edit31->Text = RoundTo(a[30], -2);
}

//-----
void __fastcall TForm1::N10Click(TObject *Sender)
{
    ShellExecute(Handle, "open", "instruction.docx", NULL, NULL,
SW_NORMAL);
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    if(Edit1->Text == "")
    {
        ShowMessage("Введіть будь-ласка радіус кривошипа");
        Edit1->SetFocus();
        return;
    }
    if(Edit2->Text == "")
    {
        ShowMessage("Введіть будь-ласа доквжину шатуна");
        Edit2->SetFocus();
        return;
    }
    if(Edit3->Text == "")

```

```

    {
        ShowMessage("Введіть будь-ласка повну довжину шатунної
шийки з урахуванням галтелей");
        Edit3->SetFocus();
        return;
    }
    if (Edit4->Text == "")
    {
        ShowMessage("Введіть будь-ласка довжину корінної
шийки");
        Edit4->SetFocus();
        return;
    }
    if (Edit5->Text == "")
    {
        ShowMessage("Введіть будь-ласка товщину щоки");
        Edit5->SetFocus();
        return;
    }
    if (Edit6->Text == "")
    {
        ShowMessage("Введіть будь-ласка ширину щоки");
        Edit6->SetFocus();
        return;
    }
    if (Edit7->Text == "")
    {
        ShowMessage("Введіть будь-ласка зовнішній діаметр
корінної шийки");
        Edit7->SetFocus();
        return;
    }
    if (Edit8->Text == "")
    {
        ShowMessage("Введіть будь-ласка внутрішній діаметр
корінної шийки");
        Edit8->SetFocus();
        return;
    }
    if (Edit9->Text == "")
    {
        ShowMessage("Введіть будь-ласка зовнішній діаметр
шатунної шийки");
        Edit9->SetFocus();
        return;
    }
    if (Edit10->Text == "")
    {
        ShowMessage("Введіть будь-ласка внутрішній діаметр
шатунної шийки");
        Edit10->SetFocus();
        return;
    }
}

```

```

        if(Edit11->Text == "")
        {
            ShowMessage("Введіть будь-ласка масу поршневої
групи");
            Edit11->SetFocus();
            return;
        }
        if(Edit12->Text == "")
        {
            ShowMessage("Введіть будь-ласка масу шатуна,
прикладену до пальця");
            Edit12->SetFocus();
            return;
        }
        if(Edit13->Text == "")
        {
            ShowMessage("Введіть будь-ласка масу шатуна,
прикладену до осі кривошипа");
            Edit13->SetFocus();
            return;
        }
        if(Edit14->Text == "")
        {
            ShowMessage("Введіть будь-ласка масу шатунної шийки");
            Edit14->SetFocus();
            return;
        }
        if(Edit15->Text == "")
        {
            ShowMessage("Введіть будь-ласка масу щоки");
            Edit15->SetFocus();
            return;
        }
        if(Edit16->Text == "")
        {
            ShowMessage("Введіть будь-ласка кут альфа");
            Edit16->SetFocus();
            return;
        }
        PageControll->ActivePage = TabSheet2;
    }
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Mem01->Clear();

    if(Edit17->Text == "")
    {
        ShowMessage("Введіть будь-ласка кутову швидкість
обертання");
        Edit17->SetFocus();
        return;
    }
}

```



```
if(Edit18->Text == "")
{
    ShowMessage("Введіть будь-ласка реакцію, яка діє на
скручування щоки");
    Edit18->SetFocus();
    return;
}
if(Edit19->Text == "")
{
    ShowMessage("Введіть будь-ласка силу тиску газу");
    Edit19->SetFocus();
    return;
}
if(Edit20->Text == "")
{
    ShowMessage("Введіть будь-ласка максимальну силу тиску
газів на днищі поршня");
    Edit20->SetFocus();
    return;
}
if(Edit21->Text == "")
{
    ShowMessage("Введіть будь-ласка коефіцієнт міцності
сталі (обертання)");
    Edit21->SetFocus();
    return;
}
if(Edit22->Text == "")
{
    ShowMessage("Введіть будь-ласка коефіцієнт міцності
сталі (вигин)");
    Edit22->SetFocus();
    return;
}
if(Edit23->Text == "")
{
    ShowMessage("Введіть будь-ласка коефіцієнт
концентрації напруг (корінна шийка)");
    Edit23->SetFocus();
    return;
}
if(Edit24->Text == "")
{
    ShowMessage("Введіть будь-ласка коефіцієнт
концентрації напруг (шатунна шийка, вигин)");
    Edit24->SetFocus();
    return;
}
if(Edit25->Text == "")
{
    ShowMessage("Введіть будь-ласка коефіцієнт
концентрації напруг (шатунна шийка, обертання)");
    Edit25->SetFocus();
}
```

```

        return;
    }
    if(Edit26->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт
концентрації напруг (шока, вигин)");
        Edit26->SetFocus();
        return;
    }
    if(Edit27->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт масштабної
чутливості (обертання)");
        Edit27->SetFocus();
        return;
    }
    if(Edit28->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт поверхневої
чутливості (обертання)");
        Edit28->SetFocus();
        return;
    }
    if(Edit29->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт масштабної
чутливості (вигин)");
        Edit29->SetFocus();
        return;
    }
    if(Edit30->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт поверхневої
чутливості (вигин)");
        Edit30->SetFocus();
        return;
    }
    if(Edit31->Text == "")
    {
        ShowMessage("Введіть будь-ласка коефіцієнт u");
        Edit31->SetFocus();
        return;
    }
    PageControll->ActivePage = TabSheet3;

    AnsiString ads;

    ads=DecimalSeparator;
    DecimalSeparator = ',';
    double
r, l, lshsh, lksh, h, b, dksh, qksh, dshsh, qshsh, mn, mshn, mshk, mshsh, msch, alpha
, w, Rr, Pr, Fn, rl, Ql, Kr, KQ, Kr1, KQ1, Emr, Enr, EmQ, EnQ, u;

```

```

r = StrToFloat(Edit1->Text);
l = StrToFloat(Edit2->Text);
lshsh = StrToFloat(Edit3->Text);
lksh = StrToFloat(Edit4->Text);
h = StrToFloat(Edit5->Text);
b = StrToFloat(Edit6->Text);
dksh = StrToFloat(Edit7->Text);
qksh = StrToFloat(Edit8->Text);
dshsh = StrToFloat(Edit9->Text);
qshsh = StrToFloat(Edit10->Text);
mn = StrToFloat(Edit11->Text);
mshn = StrToFloat(Edit12->Text);
mshk = StrToFloat(Edit13->Text);
mshsh = StrToFloat(Edit14->Text);
msch = StrToFloat(Edit15->Text);
alpha = StrToFloat(Edit16->Text);
w = StrToFloat(Edit17->Text);
Rr = StrToFloat(Edit18->Text);
Pr = StrToFloat(Edit19->Text);
Fn = StrToFloat(Edit20->Text);
rl = StrToFloat(Edit21->Text);
Ql = StrToFloat(Edit22->Text);
Kr = StrToFloat(Edit23->Text);
KQ = StrToFloat(Edit24->Text);
Kr1 = StrToFloat(Edit25->Text);
KQ1 = StrToFloat(Edit26->Text);
Emr = StrToFloat(Edit27->Text);
Enr = StrToFloat(Edit28->Text);
EmQ = StrToFloat(Edit29->Text);
EnQ = StrToFloat(Edit30->Text);
u = StrToFloat(Edit31->Text);

double iib,mj,
Pj,P,iy,al,Tmas[25],Pmas[25],ib,T,Wksh,Mkrmin,Mkrmax,rmin,rmax,rm,ra,n
r,S,K;
double pi = 3.14,Tmin=320000,Tmax=-320000,Pmin=320000,Pmax=-
320000,ar=0.1,K1=2.15,K11=1,K1sh=1.98,K11sh=0.9;
int i=iy/30;

Memol->Lines->Add("Розрахунок колінчатого вала:");

Memol->Lines->Add("");
Memol->Lines->Add("Розрахунок корінної шийки:");
mj=mn+mshn;
Memol->Lines->Add("Сумарна маса, діюча на кривошип:\t" +
FloatToStrF(mj, ffFixed, 9, 3));
try{
    al=r/l;
    for(iy=0;iy<=720;iy+=30){
        Pj=-1*mj*r*pow(w,2)*Fn*(cos(iy)+al*cos(2*iy));
        P=Pr-Pj;
        Pmas[i]=P;
        if(Pmas[i]>Pmax){

```

```

        Pmax=Pmas[i];
    }
    if(Pmas[i]<Pmin){
        Pmin=Pmas[i];
    }
    iib=al*sin(iy);
    if(iib>=-1 && iib<=1){
        ib=asin(iib);
        S=P/cos(ib);
        K=(P*cos(iy+ib))/cos(ib);
        T=(P*sin(iy+ib))/cos(ib);
        Tmas[i]=T;
        if(Tmas[i]>Tmax){
            Tmax=Tmas[i];
        }
        if(Tmas[i]<Tmin){
            Tmin=Tmas[i];
        }
    }
    else{
        continue;
    }
}
Memol->Lines->Add("Pmax:\t" + FloatToStrF(Pmax, ffFixed, 9,
3));
Memol->Lines->Add("Pmin:\t" + FloatToStrF(Pmin, ffFixed, 9,
3));
Memol->Lines->Add("Tmax:\t" + FloatToStrF(Tmax, ffFixed, 9,
3));
Memol->Lines->Add("Tmin:\t" + FloatToStrF(Tmin, ffFixed, 9,
3));
Memol->Lines->Add("Сила, діюча вздовж осі шатуна:\t" +
FloatToStrF(S, ffFixed, 9, 3));
Memol->Lines->Add("Нормальна сила, що діє в площині кривошипа
від сил газів й сил інерції мас, що рухаються поступально:\t" +
FloatToStrF(K, ffFixed, 9, 3));
Wksh=((pi*pow(dksh,3))/16)*(1-pow((qksh/dksh),4));
Memol->Lines->Add("Момент опору корінних шийок:\t" +
FloatToStrF(Wksh, ffFixed, 9, 3));
Mkrmin=Tmin*r*K1;
Memol->Lines->Add("Мінімальний скручуючий момент:\t" +
FloatToStrF(Mkrmin, ffFixed, 9, 3));
Mkrmax=Tmax*r*K1;
Memol->Lines->Add("Максимальний скручуючий момент:\t" +
FloatToStrF(Mkrmax, ffFixed, 9, 3));
rmin=Mkrmin/Wksh;
Memol->Lines->Add("Мінімальне значення дотичних напруг:\t" +
FloatToStrF(rmin, ffFixed, 9, 3));
rmax=Mkrmax/Wksh;
Memol->Lines->Add("Максимальне значення дотичних напруг:\t" +
FloatToStrF(rmax, ffFixed, 9, 3));
rm=(rmax+rmin)/2;

```

```

Memol->Lines->Add("Середня напруга:\t" + FloatToStrF(rm,
ffFixed, 9, 3));
ra=(rmax-rmin)/2;
Memol->Lines->Add("Амплітуда напруг:\t" + FloatToStrF(ra,
ffFixed, 9, 3));
nr=rl/((Kr/Emr*Enr)*ra+ar*rm);
Memol->Lines->Add("Запас міцності корінної шийки:\t" +
FloatToStrF(nr, ffFixed, 9, 3));

//-----шатунна шийка-----
double
Kmax, Kmin, Knp, Kp, Tzmax, Tzmin, RKmax, RKmin, Rrmax, Rrmin, MKmax, MKmin, l1, Mr
max, Mrmin, Mzmax, Mzmin, Wizg, Qmax, Qmin, Qa, Qm, nQ, Mshshmax, Mshshmin, Wr, rma
x1, rmin1, rml, ral, nrl, n;
double ar1=0.08, aQ=0.1;

Memol->Lines->Add(" ");
Memol->Lines->Add("Розрахунок шатунної шийки:");
Kmax=Pmax*cos(alpha+ib)/cos(ib);
Kmin=Pmin*cos(alpha+ib)/cos(ib);
Memol->Lines->Add("Kmax:\t" + FloatToStrF(Kmax, ffFixed, 9,
3));
Memol->Lines->Add("Kmin:\t" + FloatToStrF(Kmin, ffFixed, 9,
3));
Kr=(mshk+mshsh+2*msch)*r*pow(w, 2);
Memol->Lines->Add("Відцентрова сила, яка діє на шатунну
шийку:\t" + FloatToStrF(Kr, ffFixed, 9, 3));
Knp=0.7*Kr;
Memol->Lines->Add("Відцентрова сила інерції противаги:\t" +
FloatToStrF(Knp, ffFixed, 9, 3));
Kp=Kr-Knp;
Memol->Lines->Add("Розрахункова відцентрова сила:\t" +
FloatToStrF(Kp, ffFixed, 9, 3));
Tzmax=Klsh*Tmax;
Tzmin=Kl1sh*Tmin;
Memol->Lines->Add("Максимальна тангенціальна сила:\t" +
FloatToStrF(Tzmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна тангенціальна сила:\t" +
FloatToStrF(Tzmin, ffFixed, 9, 3));
RKmax=(Kmax+Kp)/2;
RKmin=(Kmin+Kp)/2;
Memol->Lines->Add("Максимальна опорна реакція в площині коліна
(RK):\t" + FloatToStrF(RKmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна опорна реакція в площині коліна
(RK):\t" + FloatToStrF(RKmin, ffFixed, 9, 3));
Rrmax=Tzmax/2;
Rrmin=Tzmin/2;
Memol->Lines->Add("Максимальна опорна реакція в площині коліна
(Rr):\t" + FloatToStrF(Rrmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна опорна реакція в площині коліна
(Rr):\t" + FloatToStrF(Rrmin, ffFixed, 9, 3));
l1=lshsh+lsh+2*h;

```

```

Memol->Lines->Add("Відстань між центрами корінних шийок:\t" +
FloatToStrF(l1, ffFixed, 9, 3));
MKmax=RKmax*l1/2;
MKmin=RKmin*l1/2;
Memol->Lines->Add("Максимальний вигин шатунної шийки в площині
коліна:\t" + FloatToStrF(MKmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальний вигин шатунної шийки в площині
коліна:\t" + FloatToStrF(MKmin, ffFixed, 9, 3));
Mrmax=Rrmax*l1/2;
Mrmin=Rrmin*l1/2;
Memol->Lines->Add("Максимальний вигин шатунної шийки в
перпендикулярній площині коліна:\t" + FloatToStrF(Mrmax, ffFixed, 9,
3));
Memol->Lines->Add("Мінімальний вигин шатунної шийки в
перпендикулярній площині коліна:\t" + FloatToStrF(Mrmin, ffFixed, 9,
3));
Mzmax=pow( (pow(MKmax,2)+pow(Mrmax,2)),1/2);
Mzmin=pow( (pow(MKmin,2)+pow(Mrmin,2)),1/2);
Memol->Lines->Add("Максимальний сумарний згинальний момент:\t"
+ FloatToStrF(Mzmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальний сумарний згинальний момент:\t"
+ FloatToStrF(Mzmin, ffFixed, 9, 3));
Wizg=(pi*pow(dshsh,3))/32*(1-pow((qshsh/dshsh),4));
Memol->Lines->Add("Wвиг:\t" + FloatToStrF(Wizg, ffFixed, 9,
3));
Qmax=Mzmax/Wizg;
Qmin=Mzmin/Wizg;
Memol->Lines->Add("Максимальна напруга від вигину:\t" +
FloatToStrF(Qmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна напруга від вигину:\t" +
FloatToStrF(Qmin, ffFixed, 9, 3));
Qa=(Qmax-Qmin)/2;
Qm=(Qmax+Qmin)/2;
nQ=Ql/((KQ/EmQ*EnQ)*Qa+aQ*Qm);
Memol->Lines->Add("Запас міцності від згинального моменту:\t"
+ FloatToStrF(nQ, ffFixed, 9, 3));
Mshshmax=Mkrmax+Rrmax*r;
Mshshmin=Mkrmin+Rrmin*r;
Memol->Lines->Add("Максимальний скручуючий момент:\t" +
FloatToStrF(Mshshmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальний скручуючий момент:\t" +
FloatToStrF(Mshshmin, ffFixed, 9, 3));
Wr=2*Wizg;
rmax1=Mkrmax/Wr;
rmin1=Mkrmin/Wr;
Memol->Lines->Add("Максимальна напруга крутіння:\t" +
FloatToStrF(rmax1, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна напруга крутіння:\t" +
FloatToStrF(rmin1, ffFixed, 9, 3));
r1=(rmax1+rmin1)/2;
ra1=(rmax1-rmin1)/2;
Memol->Lines->Add("Середня напруга:\t" + FloatToStrF(r1,
ffFixed, 9, 3));

```

```

Memol->Lines->Add("Амплітуда циклу:\t" + FloatToStrF(ral,
ffFixed, 9, 3));
nr1=r1/((Kr1/Emr*Enr)*ral+ar1*rm1);
Memol->Lines->Add("Запас міцності (обертання):\t" +
FloatToStrF(nr1, ffFixed, 9, 3));
n=nQ*nr1/pow((pow(nQ, 2)+pow(nr1, 2)),1/2);
Memol->Lines->Add("Загальний запас міцності шатунної шийки:\t"
+ FloatToStrF(n, ffFixed, 9, 3));
Memol->Lines->Add(" ");

//-----шока-----
double
a,Mizgmax,Mizgmin,Wizg1,Qizgmax,Qizgmin,c,Mlizgmax,Mlizgmin,Wlizg,Qlmax,
Qlmin,Fsch,Qsjmax,Qsjmin,Qzmax,Qzmin,Qa1,Qm1,nQ1,Mkr,Mkrmax1,Mkrmin1,
Wr1,rmax2,rmin2,rm2,ra2,nr2,n1;
double aQ1=0.18,Kr2 = 0.75,ar2 = 0.08;

Memol->Lines->Add("Розрахунок шоки:");
a=(lksh/2)+(h/2);
Mizgmax=RKmax*a;
Mizgmin=RKmin*a;

Wizg1=(b*pow(h,2))/6;
Memol->Lines->Add("Момент опору вигину:\t" +
FloatToStrF(Wizg1, ffFixed, 9, 3));
Qizgmax=Mizgmax/Wizg1;
Qizgmin=Mizgmin/Wizg1;
Memol->Lines->Add("Максимальна напруга вигину в площині
коліна:\t" + FloatToStrF(Qizgmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна напруга вигину в площині
коліна:\t" + FloatToStrF(Qizgmin, ffFixed, 9, 3));
c=r-(dshsh/2);
Mlizgmax=Rrmax*c*MKmax;
Mlizgmin=Rrmin*c*MKmin;
Wlizg=(h*pow(b,2))/6;
Qlmax=Mlizgmax/Wlizg;
Qlmin=Mlizgmin/Wlizg;
Memol->Lines->Add("Максимальна напруга вигину в площині,
перпендикулярній до коліна:\t" + FloatToStrF(Qlmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна напруга вигину в площині,
перпендикулярній до коліна:\t" + FloatToStrF(Qlmin, ffFixed, 9, 3));
Fsch=b*h;
Memol->Lines->Add("Площа шоки:\t" + FloatToStrF(Fsch, ffFixed,
9, 3));
Qsjmax=RKmax/Fsch;
Qsjmin=RKmin/Fsch;
Qzmax=pow((pow((Qizgmax+Qsjmax),2)+pow(Qlmax,2)),1/2);
Qzmin=pow((pow((Qizgmin+Qsjmin),2)+pow(Qlmin,2)),1/2);
Memol->Lines->Add("Максимальна сумарна напруга вигину:\t" +
FloatToStrF(Qzmax, ffFixed, 9, 3));
Memol->Lines->Add("Мінімальна сумарна напруга вигину:\t" +
FloatToStrF(Qzmin, ffFixed, 9, 3));
Qa1=(Qzmax-Qzmin)/2;

```

```

    Qm1=(Qzmax+Qzmin)/2;
    Mem01->Lines->Add("Амплітуда напруг:\t" + FloatToStrF(Qa1,
ffFixed, 9, 3));
    Mem01->Lines->Add("Середня напруга:\t" + FloatToStrF(Qm1,
ffFixed, 9, 3));
    nQ1=Q1/((KQ1/EmQ*EnQ)*Qa1+aQ1*Qm1);
    Mem01->Lines->Add("Запас міцності (вигин):\t" +
FloatToStrF(nQ1, ffFixed, 9, 3));

    Mkr=Rr*alpha;
    Mem01->Lines->Add("Скручування щоки під дією реакції Rr:\t" +
FloatToStrF(Mkr, ffFixed, 9, 3));
    Mkrmax1=0.5*Tzmax*a;
    Mkrmin1=0.5*Tzmin*a;
    Mem01->Lines->Add("Максимальне значення обертального
моменту:\t" + FloatToStrF(Mkrmax1, ffFixed, 9, 3));
    Mem01->Lines->Add("Мінімальне значення обертального
моменту:\t" + FloatToStrF(Mkrmin1, ffFixed, 9, 3));
    Wr1=u*b*pow(h,2);
    Mem01->Lines->Add("Момент опору скручуванню:\t" +
FloatToStrF(Wr1, ffFixed, 9, 3));
    rmax2=Mkrmax1/Wr1;
    rmin2=Mkrmin1/Wr1;
    Mem01->Lines->Add("Максимальна напруга:\t" +
FloatToStrF(rmax2, ffFixed, 9, 3));
    Mem01->Lines->Add("Мінімальна напруга:\t" + FloatToStrF(rmin2,
ffFixed, 9, 3));
    rm2=(rmax2+rmin2)/2;
    ra2=(rmax2-rmin2)/2;
    Mem01->Lines->Add("Середня напруга:\t" + FloatToStrF(rm2,
ffFixed, 9, 3));
    Mem01->Lines->Add("Амплітуда напруг:\t" + FloatToStrF(ra2,
ffFixed, 9, 3));
    nr2=r1/((Kr2/Emr*Enr)*ra2+ar2*rm2);
    Mem01->Lines->Add("Запас міцності (обертання):\t" +
FloatToStrF(nr2, ffFixed, 9, 3));
    n1=nQ1*nr2/pow((pow(nQ1, 2)+pow(nr2, 2)),1/2);

    Mem01->Lines->Add("Загальний запас міцності щоки:\t" +
FloatToStrF(n1, ffFixed, 9, 3));
    Mem01->Lines->Add(" ");
}
catch (EZeroDivide &e) {
    Mem01->Clear();
    Mem01->Lines->Add("Виникла помилка при розрахунку,
спробуйте ввести інші дані, або зв'яжіться з розробником.");
    ShowMessage("Помилка! Відбувається ділення на нуль,
перевірте правильність даних.");
}
catch(...){
    Mem01->Clear();
    Mem01->Lines->Add("Виникла помилка при розрахунку,
спробуйте ввести інші дані, або зв'яжіться з розробником.");
}

```



```

        ShowMessage("Невідома помилка!");
    }
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet1;
}
//-----

void __fastcall TForm1::Button4Click(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet2;
}
//-----

void __fastcall TForm1::Edit1KeyPress(TObject *Sender, char &Key)
{
    if((Key>='0') && (Key<='9'))
        return;
    if(Key==DecimalSeparator) {
        if(((TEdit*)Sender)->Text).Pos(DecimalSeparator)!=0)
            Key=0;
        return;
    }
    if(Key==VK_BACK)
        return;
    if(Key==VK_RETURN) {
        ((TEdit*)Sender)->SetFocus();
        return;};
    Key=0;
}

```