

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

**КВАЛІФІКАЦІЙНА РОБОТА**  
**БАКАЛАВРА**

**на тему:**

**«Інформаційна система розрахунку режимів різання  
при свердлінні»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Ободяк В.К.**

**Студента групи ІН–64–8**

**Столяр М.Ю.**

**СУМИ 2020**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
до випускної роботи**

Студента четвертого курсу, групи ІН-64-8 спеціальності “Інформатика”  
денної форми навчання Столяра Миколи Юрійовича.

**Тема: “Інформаційна система розрахунку режимів різання при свердлінні”**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2020 г.

**Зміст пояснювальної записки:** 1) аналітичний огляд існуючих рішень; 2)  
вибір інструментарію для рішення задачі; 3) виконання проектного завдання.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

Керівник випускної роботи \_\_\_\_\_ Ободяк В.К.

Завдання прийняв до виконання \_\_\_\_\_ Столяр М.Ю.

## РЕФЕРАТ

**Записка:** 58 сторінок, 16 рисунків, 8 формул, 1 додаток, 14 джерел.

**Об'єкт дослідження** — процес розрахунку режимів різання при свердлінні.

**Мета роботи** — розробка програмного додатку для розрахунку режимів різання при свердлінні.

**Методи дослідження** — метод емпіричного дослідження.

**Результати** — розроблено алгоритм та програмне забезпечення системи для обчислення режимів різання при свердлінні. Розроблений алгоритм реалізовано у формі програмного забезпечення, створеного за допомогою інструментального програмного середовища Microsoft Visual Studio.

ІНЖЕНЕРНІ РОЗРАХУНКИ, РЕЖИМИ РІЗАННЯ ПРИ  
СВЕРДЛІННІ, МАТЕРІАЛ ДЛЯ РІЗАННЯ, ПОРІВНЯЛЬНИЙ  
АНАЛІЗ.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....</b>	<b>6</b>
1.1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	6
1.2 РЕАЛІЗАЦІЇ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ.....	9
1.3 ПОСТАНОВКА ЗАДАЧІ .....	13
<b>2 ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ РІШЕННЯ ЗАДАЧІ.....</b>	<b>14</b>
2.1 ВИБІР МОВИ ПРОГРАМУВАННЯ .....	14
2.2 ВИБІР СЕРЕДОВИЩА РОЗРОБКИ .....	15
<b>3 ВИКОНАННЯ ПРОЕКТНОГО ЗАВДАННЯ .....</b>	<b>17</b>
3.1 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ .....	17
3.2 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	22
<b>ВИСНОВКИ .....</b>	<b>29</b>
<b>СПИСОК ЛІТЕРАТУРИ .....</b>	<b>30</b>
<b>ДОДАТОК.....</b>	<b>32</b>

## ВСТУП

Наявність правильних інструментів для роботи дуже важлива для інженерів, щоб вони могли виконати свою роботу. Інженерні розрахунки необхідні для всіх етапів розробки будь-яких продуктів. Маючи необхідні інструменти для цього, інженери різних галузей можуть оптимізувати процес розробки продукту, прогнозуючи поведінку і продуктивність власних проектів.

Для вирішення задачі з точних інженерних розрахунків використовується спеціальне програмне забезпечення. Яке в свою чергу спроектоване і побудоване з урахуванням потреб інженерів, з усіма інструментами і додатками для розробки продукту. Це програмне забезпечення використовує мову математичної специфікації.

Часто інженери виявляють, що велика частина їх роботи проводиться з використанням дуже схожих обчислень. Але час витрачається навіть на невеликі зміни. Використовуючи програмне забезпечення для інженерних розрахунків, ці невеликі варіації можуть бути збережені і використані повторно. Таким чином, інженери заощаджують час і можуть створювати та розробляти схови-ще найкращих практичних розрахунків. Серед головних можливостей важливе місце займають зручні засоби відладки коду, тестування та пошуку помилок, що дозволяє пришвидшити розробку продукту в цілому.

Проводячи аналіз того який вплив і значення становлять інформаційні технології для сучасного суспільства, можна зробити висновок про те, що цей вплив є надзвичайно важливим, а значення таких технологій в найближчому майбутньому буде швидко зростати.

# 1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 Огляд літературних джерел

В сучасну епоху розрахунки дуже важливі для розробки певних продуктів. Однією з основних концепцій, що відокремлює інженерів, є здатність передбачити всі характеристики перед тим, як прототип буде побудований і протестований. Крім того, інженери можуть ввести корективи, щоб привести ці характеристики були відповідні з цілями продукту. Але крім обов'язків інженерів є і наслідки для організації. Створення і тестування прототипів може бути дорогим і трудомістким. Прогнозуючі розрахунки сприяють поданню віртуального прототипу, де інженер може прогнозувати та управляти характеристиками по цілях в цифровому середовищі, а не в реальному світі.

Одна з найважливіших задач в сучасній інженерії – це розрахунок режимів різання при свердлінні [7]. Існує чотири режими різання при свердлінні [9]:

- Свердління;
- Розсвердлювання;
- Зенкування;
- Розгортання.

Свердління – це вид механічної обробки матеріалів різанням, при якому за допомогою спеціального обертового ріжучого інструменту отримують отвори різного діаметру і глибини [8].

Розсвердлювання – це процес механічної обробки свердлом наявного отвору з метою збільшення його діаметра [8].

Зенкування – це вид механічної обробки різанням, в якому за допомогою спеціальних інструментів (зенкерів) проводиться обробка циліндричних і конічних отворів в деталях з метою збільшення їх діаметра, підвищення якості поверхні і точності [8].

Розгортання – це вид різання, який призначений для механічної обробки отворів. Здійснюють після попереднього свердління та зенкування для отримання отворів високоточного розміру із гладкою поверхнею. Обертаючий

інструмент (розгортка) знімає лезами найдрібніші стружки з внутрішньої поверхні отвору [8].

Для розрахунку режиму різання необхідно обчислити такий список величин: глибина різання, подача, швидкість різання, загальний поправочний коефіцієнт, обертаючий момент, потужність різання, та частота обертання [9].

Глибина різання [9]:

- при свердлінні глибина різання  $t=0,5D$  (рис.);
- при розсвердлюванні, зенкуванні и розгортанні  $t=0,5(D-d)$ .

Подача [9]:

- при свердлінні отворів без обмежуючих факторів зазвичай обирається максимально допустима по міцності свердла.
- при розсвердлюванні отворів подача, яка рекомендована для свердління, може бути збільшена до 2 разів

Швидкість різання (м / хв) [9]:

- під час свердління

$$v = \frac{C_v D^q}{T^{m_s} S^y} K_v \quad (1.1)$$

- при розсвердлюванні, зенкуванні, розгортанні

$$v = \frac{C_v D^q}{T^{m_t} t^x S^y} K_v \quad (1.2)$$

Загальний поправочний коефіцієнт на швидкість різання, враховуючи фактичні умови різання [9]:

$$K_v = K_{mv} K_{nv} K_{lv} \quad (1.3)$$

Обертаючий момент (Н\*м): розраховують за допомогою формул [9]:

- при свердлінні

$$M_{кр} = 10C_M D^q S^y K_p; P_0 = 10C_p D^q S^y K_p \quad (1.4)$$

– при розсвердлюванні та зенкуванні

$$M_{кр} = 10C_M D^q t^x S^y K_p; P_0 = 10C_p t^x S^y K_p \quad (1.5)$$

– при розгортанні використовують формулу

$$M_{кр} = \frac{C_p t^x S_z^y D z}{2 \cdot 100} \quad (1.6)$$

Потужність різання визначається за формулою [9]:

$$N_e = \frac{M_{кр} n}{9750} \quad (1.7)$$

Частота обертання заготовки/інструменту визначається за формулою [9]:

$$n = \frac{1000v}{\pi D} \quad (1.8)$$

Перші обчислювальні комплекси створювалися для того, щоб людина могла перекласти на електронні машини величезний обсяг алгоритмізованих обчислень, тим самим позбавивши себе від ручної праці, пов'язаної з проведенням необхідних для поточної роботи розрахунків. Потім наступив бурхливий розвиток обчислювальної техніки, кібернетики і чисельних методів, що дозволило людині вирішувати складніші задачі, знаходячи оптимальніші рішення завдань, які вважалися раніше неосяжними. Обчислювальні можливості високопродуктивних систем широко використовуються в наш час при



проведенні розрахункових, проектувальних, науково-дослідних, та пошукових роботах.

Історично склалося, що спеціалізоване інженерне програмне забезпечення прийшло на зміну логарифмічній лінійці. На сьогодні інженерне програмне забезпечення дозволяє найбільш широко охопити весь етап виробничої діяльності від розробки того чи іншого механізму, до його виробництва та подальшого впровадження, це той інструмент, яким активно користується розробник при проведенні дослідницьких або розрахунково-проектувальних робіт.

Сучасне інженерне програмне забезпечення створюється з використанням популярних технологій програмування і володіє, як правило, складним математичним апаратом. Крім того, широта інтересів людини така, що неможливо створити «з нуля» повністю закінчену версію програмного продукту, яка задовольнила б усі запити користувачів. Створення дійсно потужних обчислювальних пакетів під силу лише комерційним фірмам, які співпрацюють з передовими фахівцями в галузі науки та техніки, програмування; випускаючими постійні оновлення та нові версії своїх програмних продуктів; надає технічну підтримку користувачам і здійснює навчання по роботі зі своїми програмними продуктами, що просто необхідно через складність існуючих інженерних пакетів.

Актуальність програмного забезпечення для інженерних розрахунків полягає в заощадженні спеціалістами часу при обчисленні різноманітних формул, систем алгебраїчних рівнянь, фізичних величин і т.д. Адже при виконанні певних обчислень інженер витрачає багато часу на перевірку правильності запису формул, правильності їх обчислень. Це у свою чергу знизить ризик ймовірності виникнення помилок спеціаліста до мінімуму. Програмне забезпечення для інженерних обчислень стане правильним інструментом для вирішення всіх прикладних задач, з якими зіштовхується

спеціаліст, а це сприятиме покращенню якості та пришвидшенню виконання великого обсягу роботи інженера.

## 1.2 Реалізації існуючих програмних продуктів

Існує безліч програмного забезпечення для інженерних розрахунків, проте одними з таких програм є «PTC Mathcad Prime» і «Matematica». Програма Mathcad Prime 3.0 призначена для автоматизації інженерних та математичних обчислень. Ця програма була розроблена у 2013 році [1].

Простота використання, облік одиниць виміру, інтерактивні математичні позначення, потужні функціональні можливості та відкрита архітектура цього програмного продукту надає можливість інженерам і організаціям швидше виконувати важливі процеси проектування. Сотні вбудованих математичних функцій і необмежені можливості визначення власних функцій, рішення рівнянь в чисельному та символічному вигляді, а також професійне рішення складних систем рівнянь - всі ці можливості цього додатка дозволяють проводити розширені дослідження проектно-конструкторських рішень [4].

Mathcad подає розрахунки, текст і зображення в зрозумілій формі, дозволяючи фіксувати інформацію, а також використовувати її повторно, що сприяє підвищенню якості та скорочення часу виходу продукту на ринок [4].

Можливості «Mathcad Prime 3.0» [4]:

- використовувати звичайний калькулятор для простих, повторюваних обчислень;
- обчислювати і спрощувати символічні вирази;
- використовувати для обчислення інтегралів та похідні функції;
- вирішувати системи лінійних алгебраїчних рівнянь, працювати з матрицями і визначниками;
- вирішувати системи нелінійних алгебраїчних рівнянь;
- будувати графіки як в декартових і циліндричних, так і в полярних координатах, різні діаграми та гистограми;

- створювати програми з розгалуженням і циклічними алгоритмами, використовуючи свій власний, інтуїтивно зрозумілий, мова програмування;
- вирішувати диференціальні рівняння;
- вирішувати задачі теорії ймовірності та математичної статистики;
- здійснювати обмін інформацією з іншими додатками операційної системи Windows, такими, як Excel, PowerPoint, Word;
- документувати розрахунки і створювати звітну документацію;
- має більш 600 вбудованих математичних функцій;
- підтримка шаблонів документів, форматування тексту, форматування формул;
- покращений модуль роботи з 3D-графіками;
- "математика в тексті" - можливість вводити формули безпосередньо в тексті.

PTC Mathcad Prime 3.0 включає нові можливості і пропонує користувачам в удосконаленні документування [4]:

- підтримка відповідності стандартам обчислень і документування;
- шаблони;
- математичні формули в тексті;
- форматування математичних виразів;
- копіювання та вставляння даних через буфер обміну в зовнішні додатки.

Користувацькі функції - це багаторазове використання існуючих алгоритмів, вже запрограмованих на мовах C ++, C, Fortran і т. д., а також доступне розширення стандартної функціональності PTC Mathcad.

Програма PTC Mathcad Prime 3.0 може бути доступною на таких операційних системах, як Windows 7, Windows 8, Windows 10. Також доступна опція розпаралелювання по ядрах мікропроцесора. В цій програмі доступний російський, український, англійський, французький, німецький, японський, італійський, іспанський інтерфейси.

Mathematica - це система комп'ютерної алгебри, яка широко використовується в наукових, інженерних, математичних і комп'ютерних областях. Ця система була розроблена Стівеном Вольфрамом в 1982 році [2].

Основні аналітичні можливості програми «Mathematica» [2]:

- рішення систем поліноміальних і тригонометричних рівнянь і нерівностей, а також трансцендентних рівнянь, що зводяться до них;
- обчислення зворотних рівнянь;
- спрощення виразів;
- обчислення границь;
- можливість проінтегрувати або продиференціювати функцію;
- рішення диференціальних рівнянь і рівнянь в приватних похідних;

Система також здійснює чисельні розрахунки: визначає значення з довільною точністю, здійснює поліноміальну інтерполяцію функції від довільного числа аргументів по набору відомих значень, а також розраховує ймовірності. Теоретико-числові можливості - визначення простого числа по його порядковому номеру, визначення кількості простих чисел, що не перевершують дане; дискретне перетворення Фур'є; розкладання числа на прості множники, знаходження НСД і НСК [5].

Також в систему закладені лінійно-алгебраїчні можливості — робота з матрицями (додавання, множення, знаходження оберненої матриці, множення на вектор, обчислення експоненти, розрахунок визначників), пошук власних значень і власних векторів [5].

Додаток забезпечує автоматичне генерування програмного коду мовою C і його компонування; при цьому згенеровані програми мають можливість використовуватись автономно. Для створення, обробки й оптимізації C-коду підтримується використання SymbolicC. Додатки можуть використовувати зовнішні динамічні бібліотеки, при цьому підтримується інтеграція з CUDA і OpenCL. Також Mathematica - це інтерпретована мова функціонального програмування. Можна сказати, що система Mathematica написана мовою

програмування Mathematica, хоча деякі функції, особливо ті що відносяться до лінійної алгебри, з метою оптимізації були написані мовою програмування C [3].

Mathematica підтримує і процедурне програмування із застосуванням стандартних операторів управління виконанням програми), і об'єктно-орієнтований підхід. Програма Mathematica допускає відкладені обчислення [5].

Недоліками сучасних програмних реалізацій є занадто узагальнений підхід до вирішення різноманітних інженерних задач. Це в свою чергу не дозволяє до кінця розв'язати таку вузькоспеціалізовану задачу, як розрахунок режимів різання при свердлінні. І при цьому інженер витрачає на розв'язок досить багато ресурсів. А також суттєвим недоліком сучасних програмних реалізацій є завищена ціна ліцензії. На сьогоднішній день вартість ліцензії Mathcad Prime коливається від 330 до 860 долларів США, а для програмного продукту Mathematica вартість ліцензії становить від 1010 до 2210 долларів США. Було прийнято рішення розробити програмний продукт для вузькоспеціалізованої задачі розрахунку режимів різання при свердлінні.

### **1.3 Постановка задачі**

За результати проведеного аналізу розрахунків режимів різання при свердлінні, а також існуючих програмних продуктів для інженерних розрахунків таких режимів були поставлені такі задачі:

1. вибрати мову програмування, за допомогою якої буде створено програмний продукт;
2. вибрати середовище розробки, яке б мало можливість підтримувати обрану мову програмування;
3. розробити програмне забезпечення;
4. протестувати розроблену інформаційну систему.

## 2 ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ РІШЕННЯ ЗАДАЧІ

### 2.1 Вибір мови програмування

Одним із головних аспектів при створенні будь-якого програмного додатка є вибір мови програмування. Зазвичай розробнику досить складно обрати мову програмування, за допомогою якої буде створений програмний додаток, бо нині існує дуже багато актуальних до цього часу мов програмування. Для дослідження були обрані мови програмування як: Java, C++.

Java – це об'єктно-орієнтована мова програмування, що була розроблена компанією Sun Microsystems 23 травня 1995 року. Додатки, що написані мовою програмування Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій платформі, за допомогою віртуальної Java-машини [11].

Мова Java використовується для створення інтерактивних продуктів масового застосування. В Java використовуються практично ідентичні принципи для оголошення змінних, передачі параметрів, та різноманітних операторів як і в мові програмування C++ [6].

C++ - це компільована, статично типізована мова програмування загального призначення. Мова C++ поєднує в собі такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування [10].

Мова програмування C++ має багату стандартну бібліотеку, яка містить поширені контейнери та алгоритми, введення-виведення, регулярні вирази, підтримку багатопоточності та інших можливостей. C++ поєднує властивості як високорівневих, так і низькорівневих мов. У порівнянні з його попередником - мовою Cі, - найбільшу увагу приділено підтримці об'єктно-орієнтованого та узагальненого програмування [10].

В результаті дослідження мов програмування для розробки програмного додатка було обрано мову програмування C++, адже на сьогодні це є одна з

найпопулярніших мов програмування, яка підтримується більшістю середовищами розробки. Також однією з головних переваг є доступність до навчальних матеріалів (відео уроків, статей, книжок, енциклопедій).

## 2.2 Вибір середовища розробки

Не менш важливим етапом для створення програмного додатка є вибір середовища розробки. Тому необхідно було обрати таке середовище, яке могло б підтримувати мову програмування C++.

Для пошуку необхідного середовища розробки були розглянуті декілька варіантів: Borland Builder C++, Microsoft Visual Studio, Embarcadero RAD Studio C++.

C++ Builder - це інтегроване середовище програмування яке використовується програмістами для розробки програмного забезпечення мовою програмування C++ [12].

C++ Builder містить в собі мову C++, компілятор, інтегроване середовище розробки додатків IDE (Integrated Development Environment), відладчик і різні інструменти та містить комплект загальних елементів управління, доступ до Windows API, бібліотеку візуальних компонентів VCL (Visual Component Library), компоненти та інструменти для роботи з базами даних. C++ Builder додає до процесу програмування на мові C++ можливість швидкої візуальної розробки інтерфейсу додатків [12].

Microsoft Visual Studio - це серія продуктів компанії Microsoft, що містить в собі інтегроване середовище розробки програмних продуктів та цілий широкий ряд інструментарію [13].

Visual Studio можна використовувати для створення різноманітних додатків, від простих додатків для магазину та ігор для мобільних клієнтів до великих і складних систем, які обслуговують підприємства та центри обробки даних. Засобами Visual Studio можливо створювати різноманітні програмні додатки на будь якій смак, від елементарних програм для тієї чи іншої крамниці

до масштабних систем, що здатні обслуговувати підприємства та центри обробки даних [13].

За замовчуванням Visual Studio забезпечує підтримку таких мов програмування як: C#, C і C++, JavaScript, F# і Visual Basic. Visual Studio добре працює і інтегрується зі сторонніми додатками, наприклад Unity. При цьому існує можливість розширити Visual Studio, створивши власний інструментарій для вирішення специфічних задач [13].

Embarcadero RAD Studio C++ Builder - це середовище розробки, що дозволяє створювати різноманітне програмне забезпечення, яке належить компанії Embarcadero Technologies . Інтегроване середовище розробки RAD Studio забезпечує підтримку протягом усього циклу розробки для створення єдиної бази вихідного коду, яку можна просто перекомпілювати та повторно використовувати [14].

За допомогою даного середовища розробки можна створювати програмні продукти, які зможуть ефективно працювати на таких операційних системах: Windows, Linux, MacOS, Android, iOS [14].

RAD Studio містить в собі безліч класів та інтерфейсів, що забезпечують поліморфну розробку. Тому RAD Studio пропонує розробникам вибір між двома мовами програмування (Delphi та вдосконаленим C++). Це гарантує використання передових властивостей і можливостей цих мов програмування, наприклад функції універсальних типів (для Delphi), асинхронного програмування, автоматичного підрахунку посилань (для C++) [14].

Для створення програмного додатка було обрано середовище Microsoft Visual Studio, яке має зрозумілий інтерфейс, а також є досить зручним у використанні як для професійних розробників так і для програмістів початківців. Нині середовище розробки Microsoft Visual Studio дуже популярне на IT-ринку, і є доступним для будь-якого користувача. А також має в наявності дуже великий об'єм навчальної літератури.



## 3 ВИКОНАННЯ ПРОЕКТНОГО ЗАВДАННЯ

### 3.1 Розробка програмного продукту

Програмний продукт складається з двох модулів: основного, який забезпечує основний функціонал та додаткового, що забезпечує перегляд довідкових даних (рис. 3.1).

Перехід з додаткового модуля на основний не передбачено. Блок-схема програмного продукту:

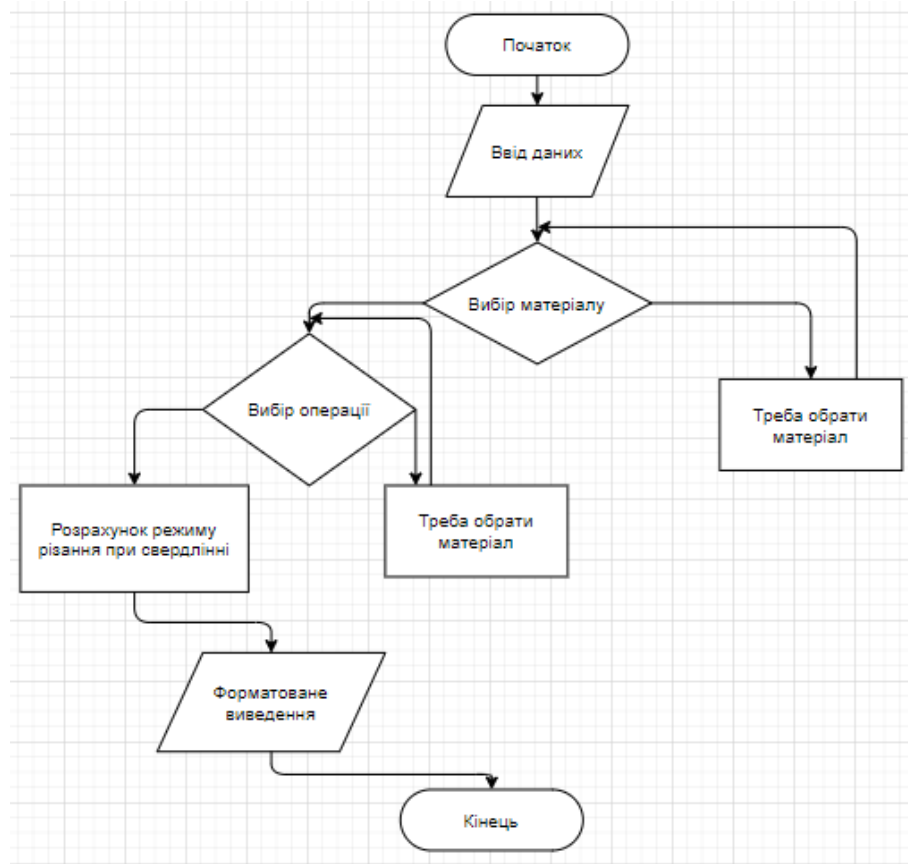


Рисунок 3.1 – Блок-схема програмного продукту

Головний модуль містить компоненти для вводу даних та інтерактивні засоби інтерфейсу. Деякі стандартні дані (режими різання) можуть бути обрані з випадального списку.

Взаємодія з користувачем реалізована такими шляхами:

- відображення результатів розрахунку у спеціальному вікні з можливістю копіювання;

– збереження результатів у файлах різного формату (\*.doc, \*.xls, \*.txt).

Для створення додатку були використані наступні компоненти:

- TextBox — поле вводу даних;
- Label — поле відображення статичних повідомлень;
- ListBox — відображення результату;
- Button — кнопка, обробник подій;
- RadioGroup — перемикач;
- ComboBox — компонент, представляє собою випадаючий список;
- GroupBox — компонент для групування елементів;
- OpenFileDialog — компонент, який забезпечує можливість відкриття файлів

з даними;

- SaveDialog — компонент, який забезпечує можливість збереження файлів з даними;

- MenuStrip — компонент для оформлення верхнього меню програми.

- ColorDialog — компонент для можливості вибору/встановлення кольору певним елементам.

- FontDialog — компонент для можливості вибору/встановлення шрифту певним елементам.

Компоненти широко використовуються в проектах значно більших масштабів. За допомогою них:

- можливо реалізувати інші особливості с формою, щоб зробити інтерфейс більш цікавим.

- програма не залежить від конкретних даних, спрацює нормально при правильному введенні початкових даних.

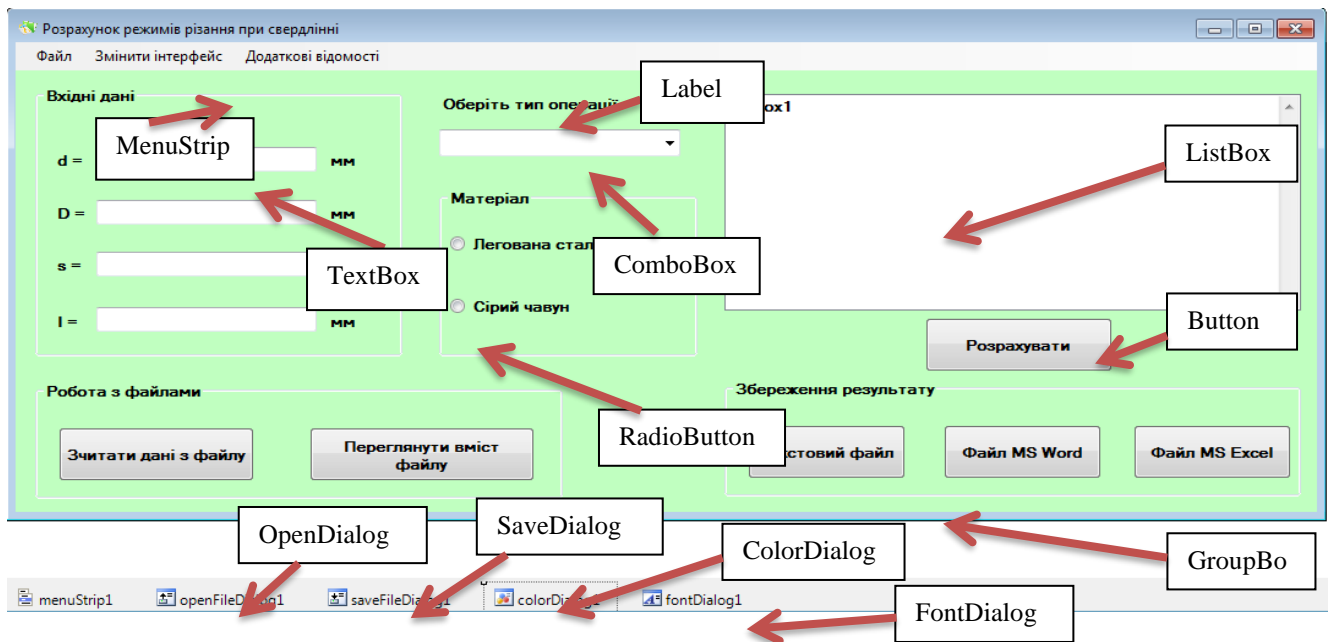


Рисунок 3.2 – Основні компоненти додатку

Для того, щоб можна було переходити з основного модуля до інших було використано такий код:

```
add_form ^form = gcnnew add_form();
form->Show();
```

Основні властивості компонентів, використані в програмному додатку:

- TextBox->Text (встановлення/визначення тексту поля вводу), Focus (встановлення фокусу на поле вводу);
- Label->Text (напис на полі виводу статичної інформації);
- ListBox->Items->Add (додавання нового рядка в поле виводу), ListBox->Items->Count (кількість заповнених рядків компонента виводу інформації);
- Button->Text (напис на кнопці);
- RadioGroup->Checked (визначення стану компонента: обраний/не обраний);
- ComboBox->SelectedIndex (номер вибраного рядка);
- OpenFileDialog->ShowDialog (відкриття діалогового вікна);
- SaveDialog->ShowDialog (відкриття діалогового вікна);
- MenuStrip->Text (напис на кнопці верхнього меню).

- ColorDialog->ShowDialog (відкриття діалогового вікна для вибору опцій), ColorDialog->Color (встановлення кольору певному компоненту);
- FontDialog->ShowDialog (відкриття діалогового вікна для вибору опцій), FontDialog->Font (встановлення шрифту певному компоненту);

Вхідними даними є виключно числові величини, введені з клавіатури. Роздільником цілої і дробової частини є кома.

У випадку не дотримання цих вимог буде відображено повідомлення про помилку (рис. 3.1, рис. 3.4).

Режими різання обираються за допомогою перемикача.

Всі поля введення даних, що розміщені на формі, є обов'язковими для заповнення. Вводити можна цілі числа та десяткові дробі.

```
d = System::Convert::ToDouble(textBox1->Text); //Ввід дробів
```

При введенні символів буде видано наступне повідомлення (рисунок 3.3):

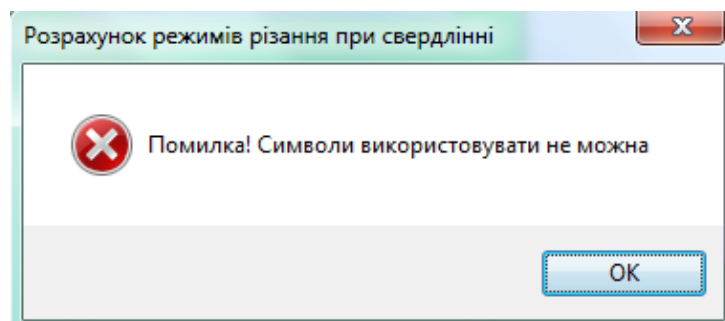


Рисунок 3.3 – Реакція програми на введення символів

В якості роздільника десяткових дробів треба використовувати кому.

```
if(Char::IsDigit(e->KeyChar) == true){}
else if(e->KeyChar == (char)Keys::Back){}
else if(e->KeyChar == (char)Keys::Enter){}
else if(e->KeyChar == ','){}
else if(e->KeyChar == '.')
{
e->Handled = true;
    MessageBox::Show("Помилка! Використовуйте в якості
    роздільника кому", "Розрахунок режимів різання при
    свердлінні",
    MessageBoxButtons::OK,
    MessageBoxIcon::Error);
}
else
```

```

{
    e->Handled = true;
    MessageBox::Show("Помилка! Символи використовувати не
можна", "Розрахунок режимів різання при свердлінні",
    MessageBoxButtons::OK, MessageBoxIcon::Error);
}

```

При введенні крапки буде видано наступне повідомлення:

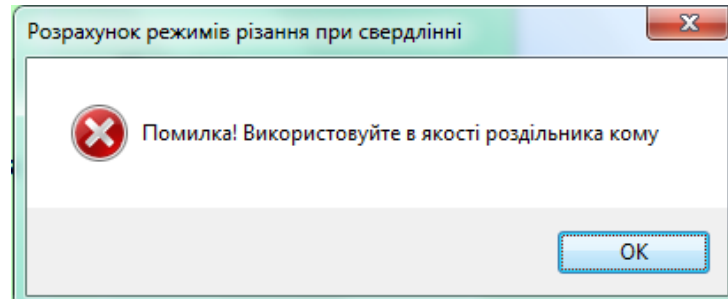


Рисунок 3.4 – Реакція програми на введення крапки

Поля Label не призначені для взаємодії з користувачем. Поле listBox також не реагує на дії користувача.

Поле Label призначене лише для виводу інформації для користувача, а поле listBox — для виводу результату.

```

else
{
    e->Handled = true;
    MessageBox::Show("Помилка! Символи використовувати не
можна", "Розрахунок режимів різання при свердлінні",
    MessageBoxButtons::OK, MessageBoxIcon::Error);
}

```

Компонент ComboBox призначений для вибору певного стану форми (за бажанням користувача).

```

if(ComboBox1->ItemIndex == 0)
{
    textBox1->Text = "1,1";
    textBox2->Text = "1,3";
    textBox3->Text = "0,5";
    textBox4->Text = "5";
    radioButton1->Checked = true;
    /*Перерахування інших компонентів*/
    /*Виконання програми*/}

```

### 3.2 Тестування програмного продукту

Мінімальні системні вимоги для встановлення програмного продукту:

- Операційна системи: Windows XP/7/8/8.1/10.
- Мінімальний об'єм місця на жорсткому диску: 80 Мб.
- Мінімальний об'єм оперативної пам'яті: 20 Мб.
- Додаткове ПЗ: Microsoft Excel 2007/2010, Microsoft Word 2007/2010.

Крок 1: запусити програму (Sverdlinnia.exe);

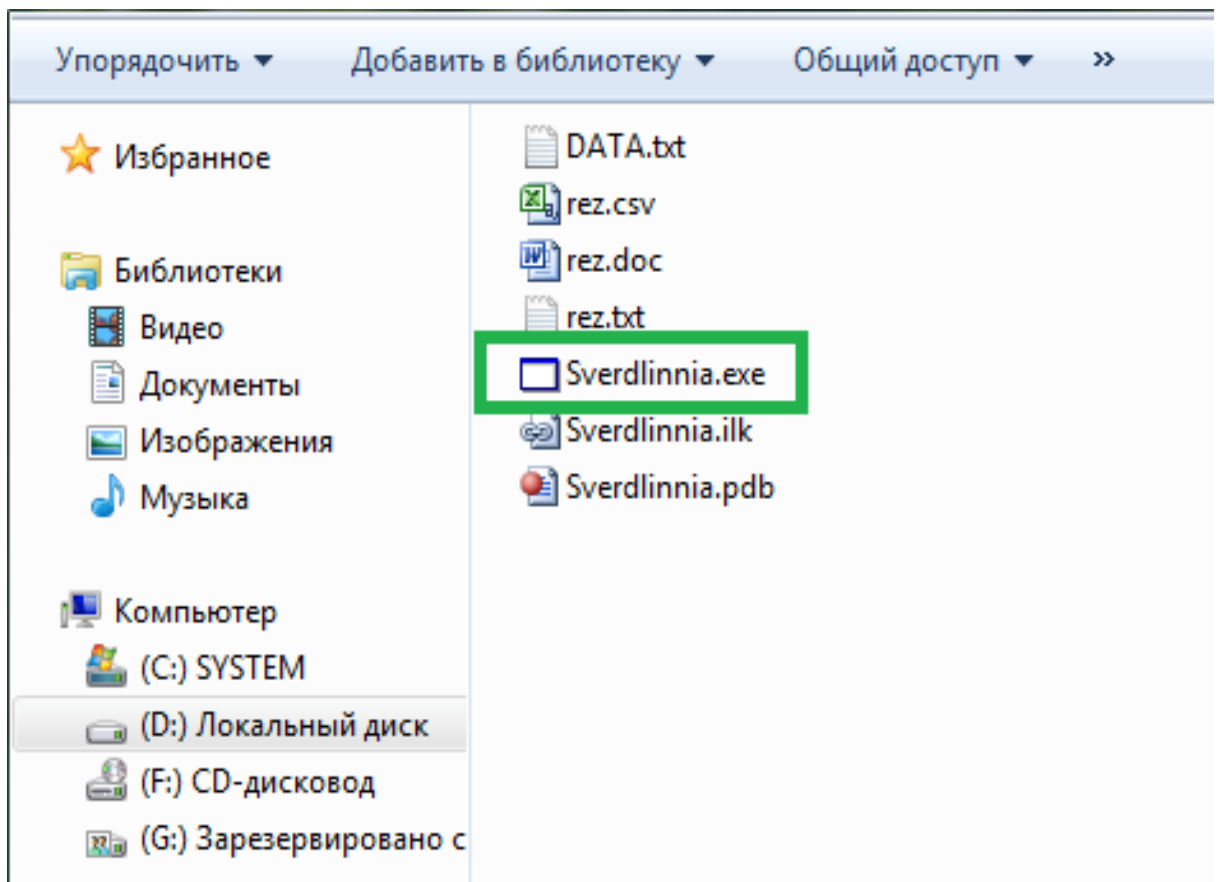


Рисунок 3.5 – Запуск програми

Крок 2: відображення початкового інтерфейсу програми;

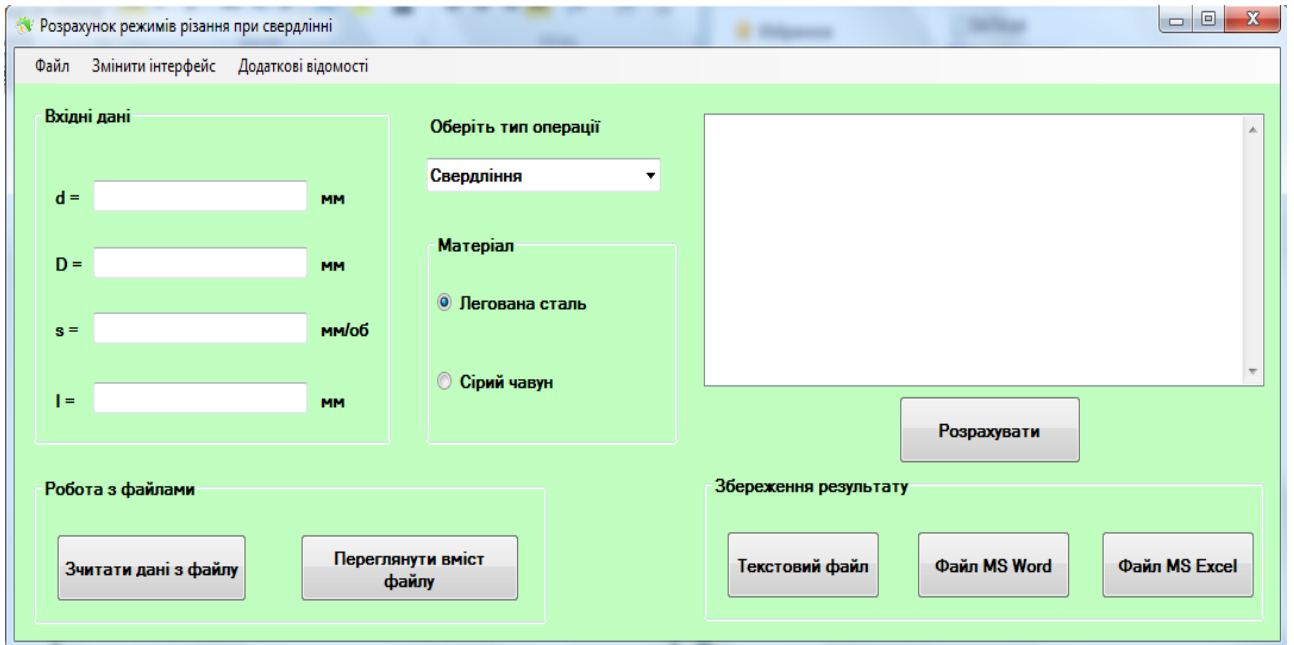


Рисунок 3.6 – Початковий інтерфейс програми

Крок 3: методи введення даних:

– зчитати з файлу;

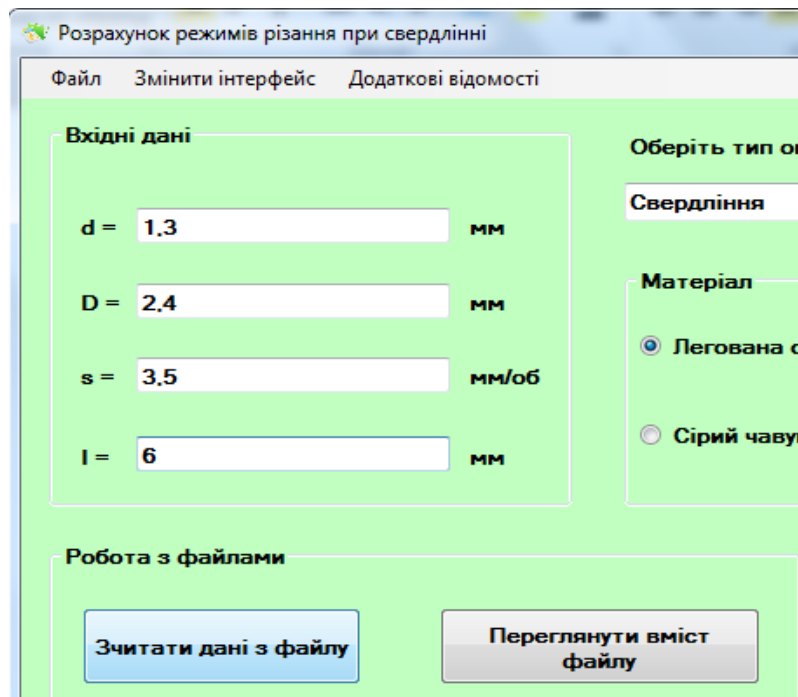


Рисунок 3.7 – Зчитування даних з файлу

– ввести вручну в поля вводу;

Рисунок 3.8 – Введення даних вручну

– виконати обчислення;

Рисунок 3.9 – Виконати розрахунок шуканих величин

Крок 4: збереження результату:

– в файл .txt;



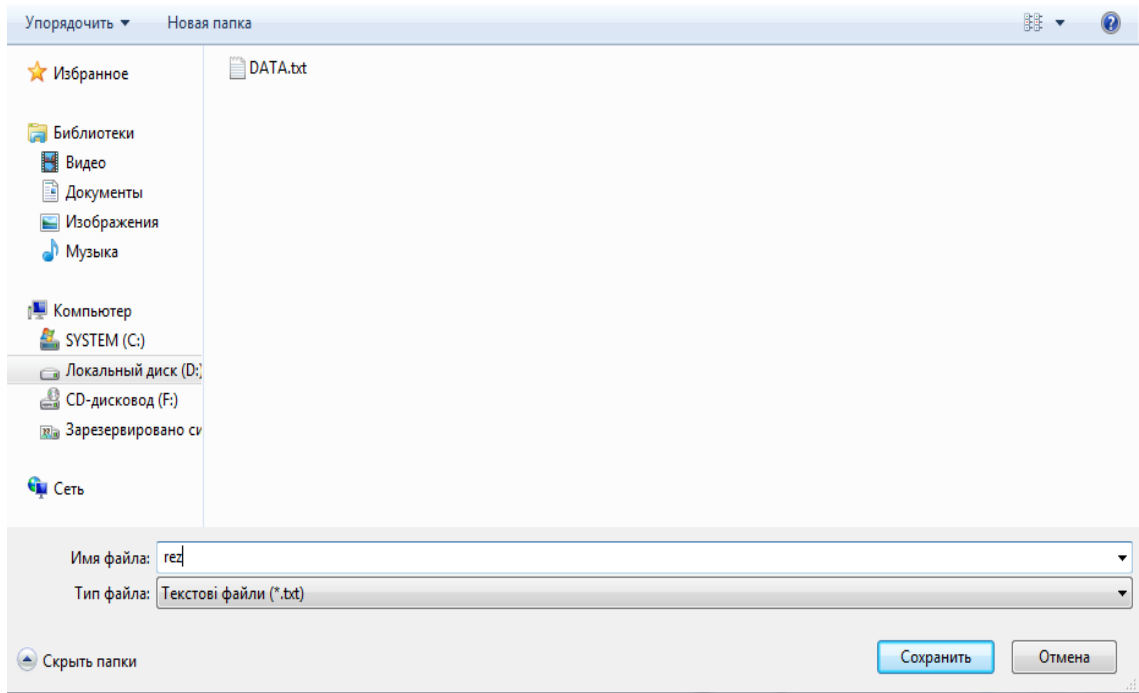


Рисунок 3.10 – Вивід даних у файл .txt

– в файл .doc;

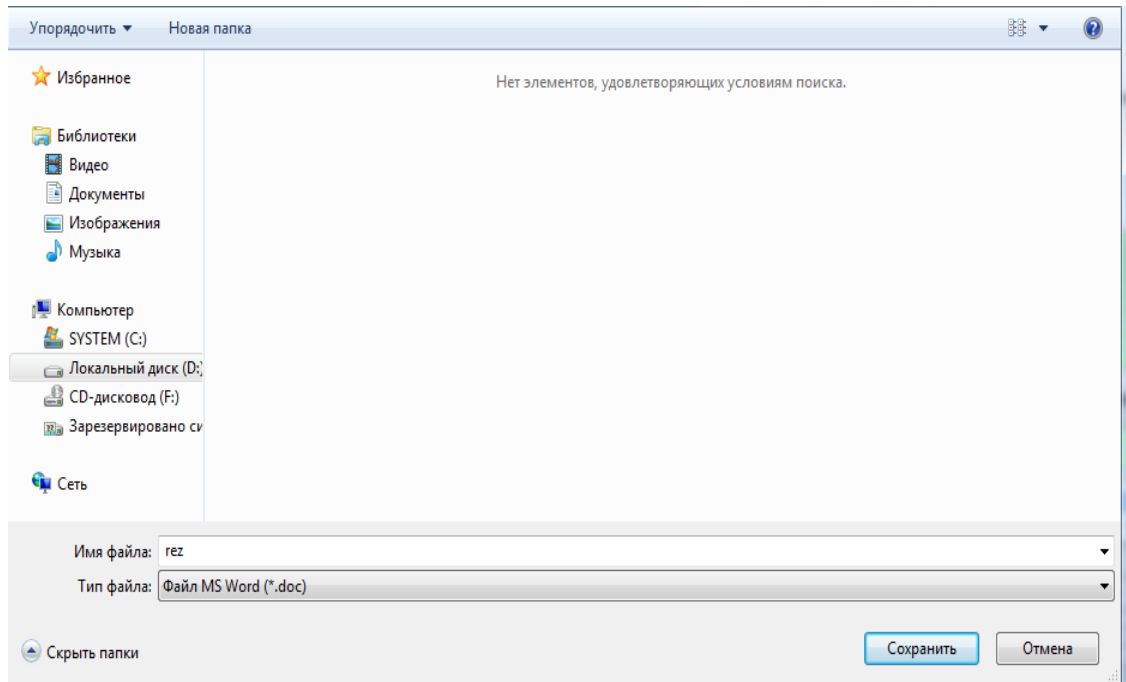


Рисунок 3.11 – Вивід даних у файл .doc

– в файл .csv;

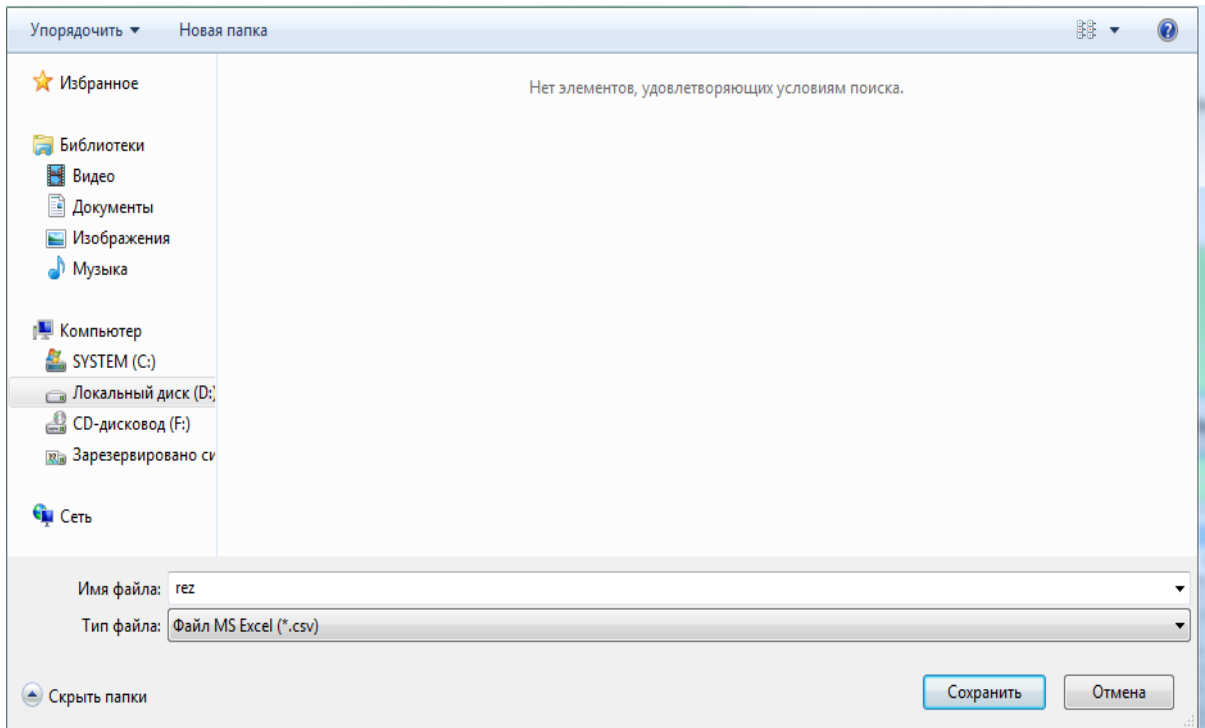


Рисунок 3.12 – Вивід даних у файл .csv

Крок 5: методи відкриття файлу для перегляду вмісту:

– переглянути файл з кінцевим результатом;

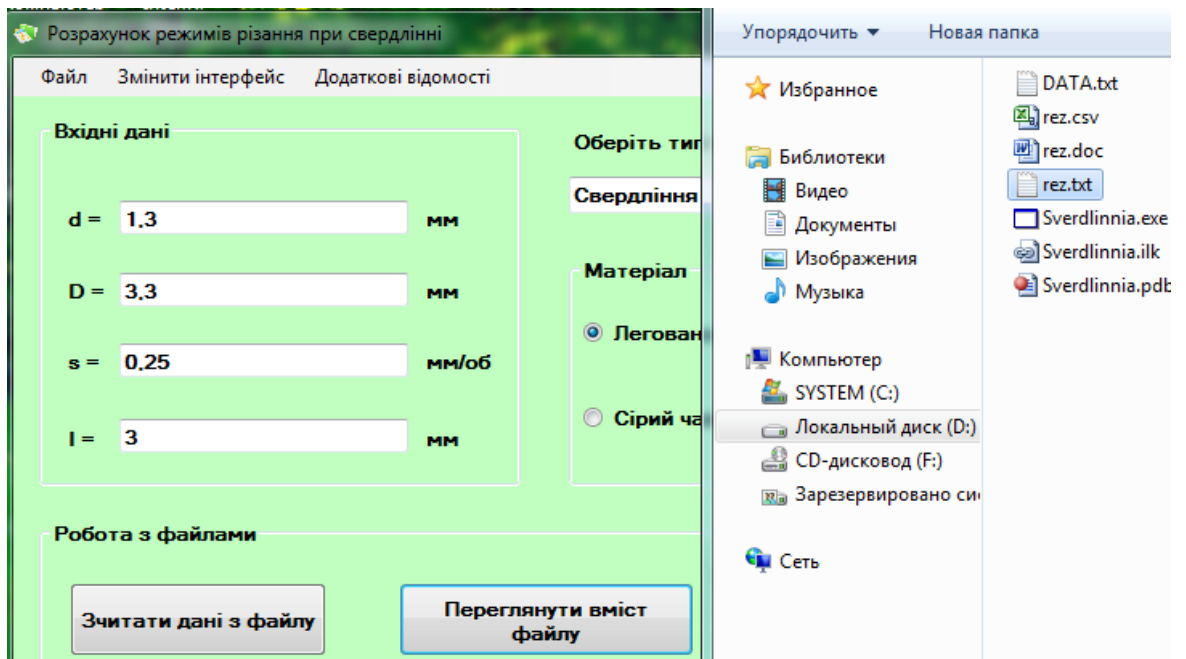


Рисунок 3.13 – Вибір файлу

```

rez.txt — Блокнот
Файл  Правка  Формат  Вид  Справка

----Результат роботи додатку----

Початкові дані:
d = 1,3 мм; D = 3,3 мм; s = 0,25 мм/об; l = 3 мм.
Виконувана операція: свердління. Матеріал: легована сталь.

Кінцеві дані:
Глибина різання: 1,65 мм.
Загальний поправочний коефіцієнт: 3,61.
Швидкість різання: 7,39 м/хв.
Частота обертання: 712,44 об/хв.
Уточнена швидкість різання: 7,39 м/хв.
Обертаючий момент: 279,45 Н*м.
Осьова сила: 161,7 Н.
Потужність різання: 20,42 кВт.
Довжина робочого ходу інструменту 15 мм.
Основний час: 0,09 хв.

```

Рисунок 3.14 – Відкриття файлу

Крок 6: другорядний модуль (відкривається через Додаткові відомості > Відомості про програму).

**Основні формули, за якими ведеться розрахунок**

$$v_p = \frac{C_v D^q}{T^m S^y} K_v,$$

$$v_p = \frac{C_v D^q}{T^m t^x S^y} K_v,$$

$$K_x = K_{sv} K_{uv} K_{lv},$$

$$n = \frac{1000 v_p}{\pi \cdot D},$$

$$v = \frac{\pi \cdot D \cdot n_{np}}{1000}, \text{ м/мин}$$

$$M_{xp} = 10 C_m D^q s^y K_p;$$

$$P_0 = 10 C_p D^q s^y K_p;$$

$$M_{xp} = 10 C_m D^q t^x s^y K_p;$$

$$P_0 = 10 C_p t^x s^y K_p;$$

$$K_p = K_{lp}.$$

$$M_{xp} = \frac{C_p t^x s^y D z}{2 \cdot 100}$$

$$N_s = \frac{M_{xp} n_{np}}{9750},$$

**Пояснення змінних**

d - початковий діаметр отвору, мм;  
D - діаметр отвору після обробки, мм;  
s - подача, мм/об;  
l - довжина оброблюваної поверхні, мм.

Рисунок 3.15 – Модуль з додатковою інформацією

Крок 7: інформація про розробника (відкривається через Додаткові відомості > Відомості про розробника).

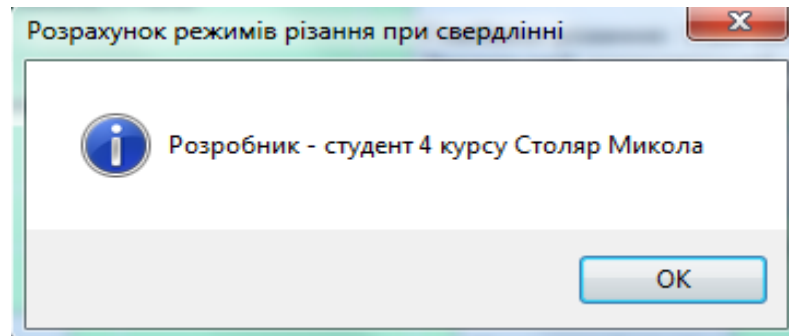


Рисунок 3.16 – Відомості про розробника

Таким чином результати тестування показують, що розроблений продукт працює коректно і доводить працездатність розробленої системи тестування.

## ВИСНОВКИ

За результатами проведеного дослідження було зроблено такі висновки.

1. Були проаналізовані сучасні програмні реалізації розрахунку режимів різання при свердлінні. Виявлено, що вони використовують узагальнений підхід до рішення різних інженерних задач, а також мають значну вартість ліцензії. Було прийняте рішення реалізувати вузькоспеціалізовану інформаційну систему для розрахунку режимів різання при свердлінні.

2. Після аналізу переваг та недоліків сучасних мов програмування для розробки програмного продукту було обрано мову програмування C++.

3. Середовищем розробки для мови програмування C++ було обрано Microsoft Visual Studio.

4. Розроблене програмне забезпечення для розрахунку режимів різання при свердлінні.

5. Тестування програмного продукту продемонструвало його надійність.

## СПИСОК ЛИТЕРАТУРИ

1. Кириянов, Д.В. Mathcad 15/ Mathcad Prime 1.0/ Д.В. Кириянов. – СПб. : «БХВ-Петербург», 2012. – 432 с.
2. Дьяконов В. П. Mathematica 5/6/7. Полный самоучитель. – ДМК Пресс, 2020. – 576 с.
3. Чарльз Генри Эдвардс, Дэвид Э. Пенни. Дифференциальные уравнения и проблема собственных значений: моделирование и вычисление с помощью Mathematica, Maple и MATLAB = Differential Equations and Boundary Value Problems: Computing and Modeling. – 3-е изд. – М.: «Вильямс», 2016. – 1104 с.
4. В.Ф. Очков, Е.П. Богомолова, Д.А. Иванов. Физико-математические этюды с Mathcad и Интернет, 2016. – 388 с.
5. Глушко В. П., Глушко А. В. Курс уравнений математической физики с использованием пакета Mathematica. – СПб.: «Лань», 2017. – 320 с.
6. Павловская, Т. А. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб. : Питер, 2017. – 461 с.
7. Горбунов Б.И. Обработка металлов резанием. – М.: Машиностроение, 1981. – 287 с.
8. Краткий справочник металлиста. / Под ред. А.Н.Малова и др. – Изд.2-е. – М.: Машиностроение, 1971. – 767 с.
9. Справочник технолога – машиностроителя. В 2 т. /Под ред. А.Г. Косиловой и Р.К. Мещерякова. – 4-е изд., перераб. и доп. – М.: Машиностроение, 1985. – 495 с.
10. Бьёрн Страуструп. Язык программирования C++ = The C++ Programming Language / Пер. с англ. – 5-е изд. – СПб.; М.: Невский диалект – Бином, 2016. – 1104 с.
11. Кей С. Хорстманн. Базовый курс = Core Java SE 9 for the Impatient. – М.: «Вильямс», 2018. – 576 с.

12. Джаррод Холингворт, Боб Сворт, Марк Кэшмэн, Поль Густавсон. Borland C++ Builder 6. Руководство разработчика = Borland C++ Builder 6 Developer's Guide. – М.: «Вильямс», 2016. – 976 с.

13. Ларс Пауерс, Майк Снелл. Visual Studio 2010 для профессионалов = Microsoft Visual Studio 2015 Unleashed. – М.: «Вильямс», 2015. – 1320 с.

14. RAD Studio 10.3.3 Now Available, Learn More [Электронный ресурс] // Sarina DuPont. – Режим доступа: <https://community.idera.com/developer-tools/b/blog/posts/rad-studio-10-3-3-now-available-learn-more>, вільний.

## ДОДАТОК

### Код програмного додатку

```
#pragma once
#include "add_form.h"
#include "stdio.h"
#include "math.h"

namespace Sverdlinnia {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Text;
using namespace System::IO;

public ref class Form1 : public System::Windows::Forms::Form
{
public:
    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: додайте код конструктора
        //
    }

protected:
    ~Form1()

```



```
{
    if (components)
    {
        delete components;
    }
}

private: System::Windows::Forms::ListBox^ listBox1;
protected:
private: System::Windows::Forms::GroupBox^ groupBox1;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::RadioButton^ radioButton2;
    private: System::Windows::Forms::RadioButton^
radioButton1;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::Button^ button7;
private: System::Windows::Forms::GroupBox^ groupBox4;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Label^ label3;
```

```
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::MenuStrip^ menuStrip1;
private: System::Windows::Forms::ToolStripMenuItem^
файлToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
змінитиІнтерфейсToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
додатковіВідомостіToolStripMenuItem;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Label^ label9;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::ToolStripMenuItem^
очиститиПоляToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
закритиToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
шрифтToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
колірToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
відомостіПроПрограмуToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^
відомостіПроРозробникаToolStripMenuItem;
    private: System::Windows::Forms::OpenFileDialog^
openFileDialog1;
    private: System::Windows::Forms::SaveFileDialog^
saveFileDialog1;
    private: System::Windows::Forms::ColorDialog^
colorDialog1;
private: System::Windows::Forms::FontDialog^ fontDialog1;
private: System::Windows::Forms::Button^ button4;
```

```

private: System::Windows::Forms::ListBox^ listBox2;

private:
    System::ComponentModel::Container ^components;
    #pragma region Windows Form Designer generated code
    void InitializeComponent(void)
    {
        #pragma region Windows Form Designer generated code
        }
    #pragma endregion

    private: System::Void
закритиToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        this->Close();
        add_form ^form = gnew add_form();
        form->Close();
    }

private: System::Void колірToolStripMenuItem_Click
(System::Object^ sender, System::EventArgs^ e)
{
    if (colorDialog1->ShowDialog() == System::Windows::Forms
::DialogResult::OK)
        this->BackColor = colorDialog1->Color;
}

private: System::Void button4_Click(System::Object^ sender,
System::EventArgs^ e){}

private: System::Void шрифтToolStripMenuItem_Click
(System::Object^ sender, System::EventArgs^ e)
{
    if (fontDialog1->ShowDialog() == System::Windows::Forms::
DialogResult::OK) {
        this->Font = fontDialog1->Font;
        groupBox1->Font = fontDialog1->Font;
        groupBox2->Font = fontDialog1->Font;
        groupBox3->Font = fontDialog1->Font;
    }
}

```

```

        groupBox4->Font = fontDialog1->Font;
        listBox1->Font = fontDialog1->Font;
        label1->Font = fontDialog1->Font;
    }}

private: System::Void очиститиПоляToolStripMenuItem_Click
(System::Object^ sender, System::EventArgs^ e)
{
    textBox1->Clear();
    textBox2->Clear();
    textBox3->Clear();
    textBox4->Clear();
    listBox1->Items->Clear();
    listBox2->Items->Clear();
    comboBox1->SelectedIndex = 0;
    radioButton1->Checked = true;
}

private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e)
{
    radioButton1->Checked = true;
    comboBox1->SelectedIndex = 0;
}

private: System::Void
відомостіПроРозробникаToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
{
    MessageBox::Show("Розробник - студент 4 курсу Столяр
Микола", "Розрахунок режимів різання при свердлінні",
MessageBoxButtons::OK, MessageBoxIcon::Asterisk);
}

private: System::Void відомостіПроПрограмуToolStripMenu
Item_Click(System::Object^ sender, System::EventArgs^ e)
{
    add_form ^form = gcnew add_form();

```

```

form->BackColor = this->BackColor;
form->groupBox1->Font = this->Font;
form->groupBox5->Font = this->Font;
form->label1->Font = this->Font;
form->label10->Font = this->Font;
form->Show();
}

private: System::Void button6_Click(System::Object^ sender,
System::EventArgs^ e)
{
openFileDialog1->Filter = "Всі файли|*.*";
openFileDialog1->Title = "Відкрити файл";
openFileDialog1->FileName = "";
if(openFileDialog1->ShowDialog() == System::Windows::Forms
::DialogResult::OK)
    System::Diagnostics::Process::Start(openFileDialog1->File
Name);
}

private: System::Void button7_Click(System::Object^ sender,
System::EventArgs^ e)
{
    openFileDialog1->Filter = "Текстові файли (*.txt)|*.txt";
openFileDialog1->Title = "Відкрити файл";
openFileDialog1->FileName = "";
int a = 0;

    if ( openFileDialog1->ShowDialog() == System::Windows::
Forms::DialogResult::OK)
{
    listBox2->Items->Clear();
    textBox1->Text = "";
    textBox2->Text = "";
    textBox3->Text = "";
}
}

```

```
textBox4->Text = "";
StreamReader^ stream = File::OpenText(openFileDialog1
    ->FileName);
while ((!stream->EndOfStream) && a<4)
{
    String^ s = stream->ReadLine();
    listBox2->Items->Add(s);
    a++;
}

if(a==0)
{
    textBox1->Focus();
}

if(a>0)
{
    if(listBox2->Items[0]!="")
    {
        textBox1->Text = listBox2->Items[0]-
            >ToString();
        textBox2->Focus();
    }
}

if(a>1)
{
    if(listBox2->Items[1]!="")
    {
        textBox2->Text = listBox2->Items[1]-
            >ToString();
        textBox3->Focus();
    }
}
```

```
    }  
    }  
  
    if(a>2)  
    {  
        if(listBox2->Items[2]!="") {  
            textBox3->Text = listBox2->Items[2]-  
                >ToString();  
            textBox4->Focus();  
        }  
    }  
  
    if(a>3)  
    {  
        if(listBox2->Items[3]!="")  
        {  
            textBox4->Text = listBox2->Items[3]-  
                >ToString();  
        }  
    }  
    }  
    }  
  
private: System::Void button1_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
    if(listBox1->Items->Count == 0)  
    {  
        MessageBox::Show("Поле виводу порожнє. Виконайте  
        розрахунок!", "Розрахунок режимів різання при  
        свердлінні", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
    else
```

```

{
    saveFileDialog1->Filter = "Текстові файли
(*.txt)|*.txt";

    saveFileDialog1->Title = "Зберегти в текстовий
файл";

    saveFileDialog1->FileName = "";

    if ( saveFileDialog1->ShowDialog() == System::Windows::
Forms::DialogResult::OK)

    {

        StreamWriter^ myWriter = gcnew
        StreamWriter(saveFileDialog1 ->FileName, true,
        System::Text::Encoding::Default);

myWriter->WriteLine("");

for(int i=0; i<listBox1->Items->Count; i++)

    {

        myWriter->WriteLine(listBox1->Items[i]->ToString());

    }

myWriter->Close();

    }

    }

private: System::Void button4_Click_1(System::Object^
sender, System::EventArgs^ e)

{

listBox1->Items->Clear();

if(textBox1->Text->Length == 0 || System::Convert::ToDouble
(textBox1->Text) <=0)

    {

        MessageBox::Show("Не заповнене поле вводу d або
значення не більше 0", "Розрахунок режимів різання
при свердлінні", MessageBoxButtons::OK,
        MessageBoxIcon::Error);

        textBox1->Focus();

    }

else if(textBox2->Text->Length == 0 || System::Convert::

```



```
ToDouble(textBox2->Text) <= 0)
{
    MessageBox::Show("Не заповнене поле вводу D або
    значення не більше 0", "Розрахунок режимів різання
    при свердлінні", MessageBoxButtons::OK,
    MessageBoxIcon::Error);

    textBox2->Focus();
}

else if(System::Convert::ToDouble(textBox1->Text) > System
::Convert::ToDouble(textBox2->Text))
{
    MessageBox::Show("Значення d має бути меншим ніж D",
    "Розрахунок режимів різання при свердлінні",
    MessageBoxButtons::OK, MessageBoxIcon::Error);

    textBox2->Focus();
}

else if(textBox3->Text->Length == 0 || System::Convert::
ToDouble(textBox3->Text) <= 0)
{
    MessageBox::Show("Не заповнене поле вводу s або
    значення не більше 0", "Розрахунок режимів різання
    при свердлінні", MessageBoxButtons::OK,
    MessageBoxIcon::Error);

    textBox3->Focus();
}

else if(textBox4->Text->Length == 0 || System::Convert
::ToDouble(textBox4->Text) <= 0)
{
    MessageBox::Show("Не заповнене поле вводу l або
    значення не більше 0", "Розрахунок режимів різання
    при свердлінні", MessageBoxButtons::OK,
    MessageBoxIcon::Error);

    textBox4->Focus();
}

else
{
```

```

double d, D, s, l, Kr, vr, v, Mkr, P0, n, Ne, T0, L,
l1 = 6, l2 = 6;
Kr = pow(220/190,0.6);
d = System::Convert::ToDouble(textBox1->Text);
D = System::Convert::ToDouble(textBox2->Text);
s = System::Convert::ToDouble(textBox3->Text);
l = System::Convert::ToDouble(textBox4->Text);
double t;
if(radioButton1->Checked == true)
{
    double Cp = 28.7, Cv = 35.4, Cm = 49.6, q =
    0.1, m = 0.8, x = 1.3, y = 0.5, S = 0.3, T =
    87, z = 6;
    double Kv, Kmv = 1.23, Knv = 1.15, Krv = 1.23;
    if(comboBox1->SelectedIndex == 0)
    {
        listBox1->Items->Add("----Результат роботи
        додатку----");
        listBox1->Items->Add("");
        listBox1->Items->Add("Початкові дані:");
        listBox1->Items->Add("");
        listBox1->Items->Add("d = " +
            ceil(d*100)/100 + " мм; D = " +
            ceil(D*100)/100 + " мм; s = " +
            ceil(s*100)/100 + " мм/об; l = " +
            ceil(l*100)/100 + " мм.");
        listBox1->Items->Add("Виконувана операція:
        свердління. Матеріал: легована сталь.");
        listBox1->Items->Add("");
        listBox1->Items->Add("Кінцеві дані:");
        listBox1->Items->Add("");
        t = D/2;
        listBox1->Items->Add("Глибина різання: " +
            ceil(t*100)/100 + " мм.");
    }
}

```

```

Kv = Kmv + Knv + Krv;
listBox1->Items->Add("Загальний поправочний
    коефіцієнт: " + ceil(Kv*100)/100 + ".");
vr = (Cv*pow(D,q)*Kv)/(pow(T,m)*pow(S,y));
listBox1->Items->Add("Швидкість різання: "
+
    ceil(vr*100)/100 + " м/хв.");
n = (1000*vr)/(3.14*D);
listBox1->Items->Add("Частота обертання: " +
    ceil(n*100)/100 + " об/хв.");
v = 3.14*D*n/1000;
listBox1->Items->Add("Уточнена швидкість рі
    зання: " + ceil(v*100)/100 + " м/хв.");
Mkr = 10*Cm*pow(D,q)*pow(s,y)*Kr;
listBox1->Items->Add("Обертаючий момент: " +
    ceil(Mkr*100)/100 + " Н*м.");
P0 = 10*Cr*pow(D,q)*pow(s,y)*Kr;
listBox1->Items->Add("Осьова сила: " +
    ceil(P0*100)/100 + " Н.");
Ne = (Mkr*n)/9750;
listBox1->Items->Add("Потужність різання: "
    + ceil(Ne*100)/100 + " кВт.");
L = l+l1+l2;
listBox1->Items->Add("Довжина робочого ходу
    інструменту " + ceil(L*100)/100 + " мм.");
T0 = L/(n*s);
listBox1->Items->Add("Основний час: " +
    ceil(T0*100)/100 + " хв.");
}
else if(comboBox1->SelectedIndex == 1)
{
    listBox1->Items->Add("----Результат роботи

```

```

        додатку----");
listBox1->Items->Add("");
listBox1->Items->Add("Початкові дані:");
listBox1->Items->Add("");
listBox1->Items->Add("d = " +
        ceil(d*100)/100 + " мм; D = " +
        ceil(D*100)/100 + " мм; s = " +
        ceil(s*100)/100 + " мм/об; l = " +
        ceil(l*100)/100 + " мм.");
listBox1->Items->Add("Виконувана операція:
розсвердлювання. Матеріал: легована
сталь.");
listBox1->Items->Add("");
listBox1->Items->Add("Кінцеві дані:");
listBox1->Items->Add("");
t = (D-d)/2;
listBox1->Items->Add("Глибина різання: " +
        ceil(t*100)/100 + " мм.");
Kv = Kmv + Knv + Krv;
listBox1->Items->Add("Загальний поправочний
коефіцієнт: " + ceil(Kv*100)/100 + ".");
vr =
(Cv*pow(D, q) *Kv) / (pow(T, m) *pow(t, x) *pow
(S, y));
listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");
n = (1000*vr) / (3.14*D);
listBox1->Items->Add("Частота обертання:
" + ceil(n*100)/100 + " об/хв.");
v = 3.14*D*n/1000;
listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");
Mkr = 10*Cm*pow(D, q) *pow(t, x) *pow(s, y) *Kr;
listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

```

```

P0 = 10*Cp*pow(t,x)*pow(s,y)*Kr;

listBox1->Items->Add("Осьова сила: " +
ceil(P0*100)/100 + " Н.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

L = l+l1+l2;

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм."); T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

else if(comboBox1->SelectedIndex == 2)
{

listBox1->Items->Add("----Результат роботи
додатку----");

listBox1->Items->Add("");

listBox1->Items->Add("Початкові дані:");

->Items->Add("");

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");

listBox1->Items->Add("Виконувана операція:
зенкування. Матеріал: легована сталь.");

listBox1->Items->Add("");

listBox1->Items->Add("Кінцеві дані:");

listBox1->Items->Add("");

t = (D-d)/2;

listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");

Kv = Kmv + Knv + Krv;

```

```

listBox1->Items->Add("Загальний
поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");

vr =
(Cv*pow(D,q)*Kv)/(pow(T,m)*pow(t,x)*pow
(S,y));

listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");

n = (1000*vr)/(3.14*D);

listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");

v = 3.14*D*n/1000;

listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");

Mkr = 10*Cm*pow(D,q)*pow(t,x)*pow(s,y)*Kr;

listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

P0 = 10*Cr*pow(t,x)*pow(s,y)*Kr;

listBox1->Items->Add("Осьова сила: " +
ceil(P0*100)/100 + " Н.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

L = l+l1+l2;

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");

T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

else if(comboBox1->SelectedIndex == 3)
{

listBox1->Items->Add("----Результат роботи
додатку----");

listBox1->Items->Add("");
}

```

```

listBox1->Items->Add("Початкові дані:");
listBox1->Items->Add("");

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");

listBox1->Items->Add("Виконувана операція:
розгортання. Матеріал: легована сталь.");
listBox1->Items->Add("");

listBox1->Items->Add("Кінцеві дані:");
listBox1->Items->Add("");

t = (D-d)/2;

listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");

Kv = Kmv + Knv + Krv;

listBox1->Items->Add("Загальний
поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");

vr =
(Cv*pow(D, q) *Kv) / (pow(T, m) *pow(t, x) *pow
(S, y));

listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");

n = (1000*vr)/(3.14*D);

listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");

v = 3.14*D*n/1000;

listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");

Mkr = (Cp*pow(t, x) * (s/z) * (D/z))/200;

listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

L = l+l1+l2;

```

```

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");

T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

}

else if(radioButton2->Checked == true)
{

double Cp = 23.5, Cv = 40.9, Cm = 31.8, q = 0,
m = 1, x = 1.2, y = 0.4, S = 0.45, T = 85, z =
4;

double Kv, Kmv = 1.18, Knv = 1, Krv = 1.03;

if(comboBox1->SelectedIndex == 0)
{

listBox1->Items->Add("----Результат роботи
додатку----");

listBox1->Items->Add("");

listBox1->Items->Add("Початкові дані:");

listBox1->Items->Add("");

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");

listBox1->Items->Add("Виконувана операція:
свердління. Матеріал: сірий чавун.");

listBox1->Items->Add("");

listBox1->Items->Add("Кінцеві дані:");

listBox1->Items->Add("");

t = D/2;

listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");

Kv = Kmv + Knv + Krv;

```



```

listBox1->Items->Add("Загальний
поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");

vr = (Cv*pow(D,q)*Kv)/(pow(T,m)*pow(S,y));

listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");

n = (1000*vr)/(3.14*D);

listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");

v = 3.14*D*n/1000;

listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");

Mkr = 10*Cm*pow(D,q)*pow(s,y)*Kr;

listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

P0 = 10*Ср*pow(D,q)*pow(s,y)*Kr;

listBox1->Items->Add("Осьова сила: " +
ceil(P0*100)/100 + " Н.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

L = l+l1+l2;

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");

T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

else if(comboBox1->SelectedIndex == 1)
{

listBox1->Items->Add("----Результат роботи
додатку----");

listBox1->Items->Add("");

listBox1->Items->Add("Початкові дані:");

listBox1->Items->Add("");
}

```

```

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");

listBox1->Items->Add("Виконувана операція:
розсвердлювання. Матеріал: сірий чавун.");

listBox1->Items->Add("");

listBox1->Items->Add("Кінцеві дані:");

listBox1->Items->Add("");

t = (D-d)/2;

listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");

Kv = KmV + Knv + Krv;

listBox1->Items->Add("Загальний
поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");

vr =
(Cv*pow(D, q) *Kv) / (pow(T, m) *pow(t, x) *pow
(S, y));

listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");

n = (1000*vr)/(3.14*D);

listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");

v = 3.14*D*n/1000;

listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");

Mkr = 10*Cm*pow(D, q) *pow(t, x) *pow(s, y) *Kr;

listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

P0 = 10*Cr*pow(t, x) *pow(s, y) *Kr;

listBox1->Items->Add("Осьова сила: " +
ceil(P0*100)/100 + " Н.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

```

```

L = l+l1+l2;

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");

T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

else if(comboBox1->SelectedIndex == 2)
{

listBox1->Items->Add("----Результат роботи
дodatку----");

listBox1->Items->Add("");

listBox1->Items->Add("Початкові дані:");

listBox1->Items->Add("");

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");

listBox1->Items->Add("Виконувана операція:
зенкування. Матеріал: сірий чавун.");

listBox1->Items->Add("");

listBox1->Items->Add("Кінцеві дані:");

listBox1->Items->Add("");

t = (D-d)/2;

listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");

Kv = Kmv + Knv + Krv;

listBox1->Items->Add("Загальний
поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");

vr =
(Cv*pow(D, q) *Kv) / (pow(T, m) *pow(t, x) *pow
(S, y));

listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");

```

```

n = (1000*vr)/(3.14*D);

listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");

v = 3.14*D*n/1000;

listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");

Mkr = 10*Cm*pow(D,q)*pow(t,x)*pow(s,y)*Kr;

listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");

P0 = 10*Cp*pow(t,x)*pow(s,y)*Kr;

listBox1->Items->Add("Осьова сила: " +
ceil(P0*100)/100 + " Н.");

Ne = (Mkr*n)/9750;

listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");

L = l+l1+l2;

listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");

T0 = L/(n*s);

listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

}

else if(comboBox1->SelectedIndex == 3)

{

listBox1->Items->Add("----Результат роботи
додатку----");

listBox1->Items->Add("");

listBox1->Items->Add("Початкові дані:");

listBox1->Items->Add("");

listBox1->Items->Add("d = " +
ceil(d*100)/100 + " мм; D = " +
ceil(D*100)/100 + " мм; s = " +
ceil(s*100)/100 + " мм/об; l = " +
ceil(l*100)/100 + " мм.");
}

```

```

listBox1->Items->Add("Виконувана операція: розгортання. Матеріал: сірий чавун.");
listBox1->Items->Add("");
listBox1->Items->Add("Кінцеві дані:");
listBox1->Items->Add("");
t = (D-d)/2;
listBox1->Items->Add("Глибина різання: " +
ceil(t*100)/100 + " мм.");
Kv = Kmv + Knv + Krv;
listBox1->Items->Add("Загальний поправочний коефіцієнт: " +
ceil(Kv*100)/100 + ".");
vr =
(Cv*pow(D, q) *Kv) / (pow(T, m) *pow(t, x) *pow
(S, y));
listBox1->Items->Add("Швидкість різання: "
+ ceil(vr*100)/100 + " м/хв.");
n = (1000*vr) / (3.14*D);
listBox1->Items->Add("Частота обертання: "
+ ceil(n*100)/100 + " об/хв.");
v = 3.14*D*n/1000;
listBox1->Items->Add("Уточнена швидкість
різання: " + ceil(v*100)/100 + " м/хв.");
Mkr = (Cp*pow(t, x) *(s/z) *(D/z)) /200;
listBox1->Items->Add("Обертаючий момент: "
+ ceil(Mkr*100)/100 + " Н*м.");
Ne = (Mkr*n) /9750;
listBox1->Items->Add("Потужність різання:
" + ceil(Ne*100)/100 + " кВт.");
L = l+l1+l2;
listBox1->Items->Add("Довжина робочого
ходу інструменту " + ceil(L*100)/100 + "
мм.");
T0 = L / (n*s);
listBox1->Items->Add("Основний час: " +
ceil(T0*100)/100 + " хв.");

```

```

        }
    }
}

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e)
{
    if(listBox1->Items->Count == 0)
    {
        MessageBox::Show("Поле виводу порожнє. Виконайте
        розрахунок!", "Розрахунок режимів різання при
        свердлінні", MessageBoxButtons::OK,
        MessageBoxIcon::Error);
    }
    else
    {
        saveFileDialog1->Filter = "Файл MS Word (*.doc)|*.
        doc";

        saveFileDialog1->Title = "Зберегти в файл MS Word";
        saveFileDialog1->FileName = "";

        if ( saveFileDialog1->ShowDialog() ==
        System::Windows ::Forms::DialogResult::OK)
        {
            StreamWriter^ myWriter = gnew
            StreamWriter(saveFileDialog1->FileName, true,
            System::Text::Encoding:: Default);

            myWriter->WriteLine("");
            for(int i=0; i<listBox1->Items->Count; i++)
            {
                myWriter->WriteLine(listBox1->Items[i]-
                >ToString());
            }
            myWriter->Close();
        }
    }
}

```

```

}

private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e)

{

if(listBox1->Items->Count == 0)

{

    MessageBox::Show("Поле виводу порожнє. Виконайте
    розрахунок!", "Розрахунок режимів різання при
    свердлінні", MessageBoxButtons::OK,
    MessageBoxIcon::Error);

}

else

{

    saveFileDialog1->Filter = "Файл MS Excel (*.csv)|*
    .csv";

    saveFileDialog1->Title = "Зберегти в файл MS Excel";

saveFileDialog1->FileName = "";

    if ( saveFileDialog1->ShowDialog() ==
    System::Windows ::Forms::DialogResult::OK)

    {

        StreamWriter^ myWriter = gnew
        StreamWriter(saveFileDialog1->FileName, true,
        System::Text::Encoding::Default);

myWriter->WriteLine("");

for(int i=0; i<listBox1->Items->Count; i++)

{

    myWriter->WriteLine(listBox1->Items[i]-
    >ToString());

}

myWriter->Close();

}

}

}

private: System::Void textBox1_KeyPress(System::Object^
sender, System::Windows::Forms::KeyPressEventArgs^ e)

```

```

{
if(Char::IsDigit(e->KeyChar) == true){}
else if(e->KeyChar == (char)Keys::Back){}
else if(e->KeyChar == (char)Keys::Enter){}
else if(e->KeyChar == ','){}
else if(e->KeyChar == '.')
{
    e->Handled = true;

    MessageBox::Show("Помилка! Використовуйте в якості
роздільника кому", "Розрахунок режимів різання при
свердлінні", MessageBoxButtons::OK,
MessageBoxIcon::Error);
}
else
{
    e->Handled = true;

    MessageBox::Show("Помилка! Символи використовувати
не можна", "Розрахунок режимів різання при
свердлінні", MessageBoxButtons::OK,
MessageBoxIcon::Error);
}
}

private: System::Void textBox2_KeyPress(System::Object^
sender, System::Windows::Forms::KeyPressEventArgs^ e)
{
if(Char::IsDigit(e->KeyChar) == true){}
else if(e->KeyChar == (char)Keys::Back){}
else if(e->KeyChar == (char)Keys::Enter){}
else if(e->KeyChar == ','){}
else if(e->KeyChar == '.')
{
    e->Handled = true;

    MessageBox::Show("Помилка! Використовуйте в якості
роздільника кому", "Розрахунок режимів різання при

```



```
        свердлінні", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
else  
{  
    e->Handled = true;  
    MessageBox::Show("Помилка! Символи використовувати  
не можна", "Розрахунок режимів різання при  
свердлінні", MessageBoxButtons::OK,  
    MessageBoxIcon::Error);  
}  
}  
  
private: System::Void textBox3_KeyPress(System::Object^  
sender, System::Windows::Forms::KeyPressEventArgs^ e)  
{  
    if(Char::IsDigit(e->KeyChar) == true){}  
    else if(e->KeyChar == (char)Keys::Back){}  
    else if(e->KeyChar == (char)Keys::Enter){}  
    else if(e->KeyChar == ','){}  
    else if(e->KeyChar == '.')  
    {  
        e->Handled = true;  
        MessageBox::Show("Помилка! Використовуйте в якості  
роздільника кому", "Розрахунок режимів різання при  
свердлінні", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
else  
{  
    e->Handled = true;  
    MessageBox::Show("Помилка! Символи використовувати  
не можна", "Розрахунок режимів різання при  
свердлінні", MessageBoxButtons::OK,  
    MessageBoxIcon::Error);  
}  
}
```

```
}  
  
private: System::Void textBox4_KeyPress(System::Object^  
sender, System::Windows::Forms::KeyPressEventArgs^ e)  
{  
    if(Char::IsDigit(e->KeyChar) == true){}  
    else if(e->KeyChar == (char)Keys::Back){}  
    else if(e->KeyChar == (char)Keys::Enter){}  
    else if(e->KeyChar == ','){}  
    else if(e->KeyChar == '.')  
    {  
        e->Handled = true;  
        MessageBox::Show("Помилка! Використовуйте в  
        якості роздільника кому", "Розрахунок режимів  
        різання при свердлінні", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
    else  
    {  
        e->Handled = true;  
        MessageBox::Show("Помилка! Символи  
        використовувати не можна", "Розрахунок режимів  
        різання при свердлінні", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
}  
};  
}
```