



- [2] Mudra I., Kuharska O. Chat-boty yak instrument dlya populyaryzatsiyi materialiv ZMI. Visnyk Natsional'noho universytetu «Lvivska politekhnik». № 2 (2), 2021. S.69-75.
- [3] Nalyvayko O., & Malyutina A. (2021). Vykorystannya chat botiv v osvith'omu protsesi zakladu vyshchoyi osvity. Naukovi zapysky kafedry pedahohiky, 1(48), 117-122. <https://doi.org/10.26565/2074-8167-2021-48-14>
- [4] Prohnoz brytanskoyi analitychnoyi kompaniyi Teghnavio: web-site. URL: <https://worldvision.com.ua/rynok-uslug-na-baze-iskusstvennogo-intellekta-budet-ezhegodno-rasti-na-47/2>
- [5] Trofymenko O. H. Sfery zastosuvannya chat-botiv. Informatsiyne suspil'stvo: problemy ta perspektyvy : materialy VII Vseukrayins'koyi naukovo-praktychnoyi konferentsiyi (m. Odesa, 20 travnya 2022 r.) / vidp. red. N. I. Lohinova. Odesa, 2022. S. 68-71.
- [6] A Survey of Design Techniques for Conversational Agents. K. Ramesh, S. Ravishankaran, A. Joshi, K. Chandrasekaran. Communications in Computer and Information Science. 2017. URL: https://doi.org/10.1007/978-981-10-6544-6_31.
- [7] Pesonen J. 'Are You OK?' Students' Trust in a Chatbot Providing Support Opportunities. *International Conference on Human-Computer Interaction HCII 2021: Learning and Collaboration Technologies: Games and Virtual Environments for Learning*. 2021. P. 199-215. DOI: 10.1007/978-3-030-77943-6_13.
- [8] Shevat A. Designing Bots: Creating Conversational Experiences. Sebastopol. 2018. URL: https://www.accenture.com/t20180509T102140Z_w_us-en/acnmedia/PDF77/AccentureResearch-Conversational-AI-Platforms.pdf
- [9] (2022, 25 December) TZ Na rozrobku chat-bota [TT for chat-bot development] from <https://avada-media.ua/ua/services/tz-dlya-chat-bota/> [In Ukraine]
- [10] McVicker D. (2023, 10 January) 6 Common Chatbot Mistakes Your Team Might Be Making from <https://onlim.com/en/6-common-chatbot-mistakes-your-team-might-be-making/> [In English]
- [11] (2023, 10 January) Chatbot Testing: Features to Check & Quality Tips to Keep In Mind from <https://www.qamadness.com/chatbot-testing/> [In English]

Отримана в редакції 20.04.2023. Прийнята до друку 22.05.2023. Received 20 April 2023. Approved 22 May 2023. Available in Internet 19 June 2023.

УДК 681-5; 004.4

РОЗРАХУНОК КОНІЧНОГО З'ЄДНАННЯ ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ C#

Гулієва Н.М.¹, Батюк Д.Р.², Грибок Н.О.³, Гулієва З.Н.⁴

^{1,2,3} Луцький національний технічний університет, м. Луцьк, Україна

⁴ Державний податковий університет, м. Ірпінь, Україна

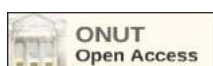
ORCID: ¹<https://orcid.org/0000-0001-9282-4880>

E-mail: ¹n.huliieva@lntu.edu.ua, ²demanbatuk@gmail.com, ³nazarjj12345@gmail.com, ⁴zemfira.huliieva.@gmail.com

Copyright © 2021 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v15i2.2526

Анотація. На сьогоднішній день розробка програмного забезпечення є актуальним видом діяльності та застосовується практично у всіх сферах суспільного життя. Програмне забезпечення виконує роль пришвидшення і покращення технологічних процесів на підприємстві, що в свою чергу призводить до виготовлення більшої кількості продукції та підвищення її якості. Неякісне програмне забезпечення або його відсутність призводить до додаткових затрат. Це в свою чергу призводить до подорожчання кінцевого продукту, що негативно впливає на виробництво й, зокрема, на ВВП економіки країни.

У науковій роботі розроблене програмне забезпечення в інтегрованому середовищі Microsoft Visual Studio 2022 для розрахунку кінцевого з'єднання на мові програмування C#. Оскільки кінчне з'єднання має складну будову таких технічних параметрів як: герметичність, висока міцність, можливість легкого регулювання



допуску та натягу за допомогою зміни осьового розташування деталей, здатність конічної пари до швидкого розбирання та збирання, самоцентрованість, то воно потребує точного і правильного розрахунку. Конічні з'єднання мають хороші технологічні властивості і широко застосовуються в промисловості, але при цьому мають відхилення від номінального конуса (діаметру, довжини, кута). Для нормальної експлуатації необхідно, щоб відхилення дійсних розмірів конуса знаходилися у межах заданих допусків. Допуски та посадки для конічних з'єднань встановлено в ДСТУ 2499-94. Конічні з'єднання характеризуються конічною посадкою та базовою відстанню з'єднання, які потрібно розрахувати, що і розглядається в даній роботі, а саме розробка інтерфейсу для розрахунку конічного з'єднання в інтегрованому середовищі Microsoft Visual Studio 2022 на мові програмування C#. Даний інтерфейс простий та зручний у користуванні та пришвидшує процес розрахунку на відміну від традиційного.

Abstract. Today, software development is a relevant activity and is used in almost all spheres of social life. Software performs the role of speeding up and improving technological processes at the enterprise, which in turn leads to the production of more products and their quality improvement. Low-quality software or its absence leads to additional costs. This, in turn, leads to an increase in the price of the final product, which negatively affects production and, in particular, the GDP of the country's economy.

In the scientific work, software was developed in the Microsoft Visual Studio 2022 integrated environment for the calculation of a conical connection in the C# programming language. Since the conical connection has a complex structure of such technical parameters as: tightness, high strength, the possibility of easy adjustment of the tolerance and tension by changing the axial location of the parts, the ability of the conical pair to be quickly disassembled and assembled, self-centering, it requires an accurate and correct calculation. Conical joints have good technological properties and are widely used in industry, but at the same time they have deviations from the nominal cone (diameter, length, angle). For normal operation, it is necessary that the deviations of the actual dimensions of the cone are within the specified tolerances. Tolerances and fits for conical joints are established in DSTU 2499-94. Taper joints are characterized by a taper fit and the base distance of the connection, which must be calculated, which is considered in this paper, namely, the development of an interface for calculating a taper connection in the integrated environment of Microsoft Visual Studio 2022 in the C# programming language. This interface is simple and convenient to use and speeds up the calculation process, unlike the traditional one.

Ключові слова: інтегроване середовище Microsoft Visual Studio 2022, мова програмування C#, програмний продукт, конічне з'єднання.

Keywords: Microsoft Visual Studio 2022 integrated environment, C# programming language, software product, conical connection.

Вступ

В нинішніх умовах розробка програмного забезпечення перетворилася на одну з найдорожчих і найзатребуваних індустрій [1 с. 12]. Будь-які вузькі місця, які знаходяться в технологічному процесі потребують написання програмного забезпечення, а його відсутність може призвести до небажаних результатів. Подовження термінів розробки, загрожує подорожчанням кінцевого продукту, а не виявлені в ході тестування помилки можуть призвести до зниження продуктивності [2 с. 22]. Примітивні помилки, невизначені повідомлення та неохайний інтерфейс не подобаються користувачам, які внаслідок цього вибирають якісніший продукт конкурента, а підприємство ризикує втратити не тільки споживачів, а й свою частку ринку. Тому, якість програмного забезпечення набуває важливого значення.

Актуальність теми наукової роботи полягає у тому, що розробка прикладних програм відіграє важливу роль у різних сферах промисловості. Дозволяє здобувати навички проектування прикладних програм для розрахунку конічного з'єднання на мові програмування C# в інтегрованому середовищі Microsoft Visual Studio 2022.

Аналіз літературних даних і постановка проблеми

Аналіз останніх досліджень і публікацій свідчить про підвищену увагу до розробок програмного забезпечення для проектування технологічного процесу прикладних програм. Зокрема, розробка та проектування програмного забезпечення відображено у працях G. Büyüközkan, S. Gülergüz [3], A. В. Рубана, В. В. Пастернак, Н. М. Гулієвої, [4] Я. П. Замора [5], І. А. Андреева, І.О. Мікульонка [6], X. Franch, J. P. Carvallo [7], R. Garg, R. Sharma, K. Sharma [8] та інших.

Мета і завдання дослідження

Метою моєї наукової роботи є розробка програмного забезпечення для розрахунку конічного з'єднання на мові програмування C# в інтегрованому середовищі Microsoft Visual Studio 2022.

Завданнями розробки програмного забезпечення є полегшення розрахунку конічного з'єднання. Це програмне забезпечення прискорює процес розрахунку та призводить до виведення точніших вихідних даних порівняно з звичайним розрахунком.

Методи і матеріали досліджень

Конічні з'єднання – це з'єднання зовнішнього та внутрішнього конусів, що мають однакові кути конуса або однакові конусності (рис. 1). Конічні сполуки в порівнянні з циліндричними мають ряд переваг і в деяких випадках є незамінними. Вони можуть бути рухомими, нерухомими та щільними [9 с. 52].

Конічні з'єднання широко застосовуються у машинах, приладах, апаратах, трубопроводах. На якість конічних з'єднань впливають похибки кутів і відхилення форми поверхонь, що сполучаються. Для підвищення точності



центрування, здатності навантаження, зносостійкості і герметичності з'єднань необхідно забезпечити рівномірний контакт поверхонь, що сполучаються.

Конічні з'єднання служать для передачі обертальних моментів, центрування деталей, що з'єднуються, регулювання в процесі складання величини зазору або натягу шляхом взаємних осьових зсувів деталей, що з'єднуються, і ущільнення стиків. Аналогічно циліндричним з'єднанням вони поділяються на три різновиди: рухомі з'єднання з гарантованим допуском.

Ці з'єднання широко застосовуються у машинах, приладах, апаратах, трубопроводах. На якість конічних з'єднань впливають похибки кутів і відхилення форми поверхонь, що сполучаються. Для підвищення точності центрування, здатності навантаження, зносостійкості і герметичності з'єднань необхідно забезпечувати рівномірний контакт поверхонь, що сполучаються. Однак це дуже трудомістка операція і при ній порушується взаємозамінність парних конусів, тому взаємне притирання застосовують тільки при дуже високих вимогах до точності та герметичності з'єднань. Дані з'єднання в порівнянні з циліндричними забезпечують можливість більшої кількості повторних збірок, а їх розрахунки аналогічні розрахункам циліндричних з'єднань. Для сучасного виробництва вони вважаються дуже перспективними.

Конічне з'єднання характеризується конічною посадкою та базовою відстанню цього з'єднання. Для конічних з'єднань встановлені посадки з натягом, допуском та перехідними даними. Величина одержуваного допуску або натягу визначається різницею діаметрів внутрішнього і зовнішнього конусів в їх поперечних перерізах, що поєднуються після фіксації взаємного осьового положення деталей, що сполучаються.

Переваги конічних з'єднань полягають у забезпеченні необхідної посадки, центрування, легкості розбирання.

Виконаємо розрахунок конічного з'єднання для метричних одиниць де втулка запресовується в кінець валу (рис. 2).

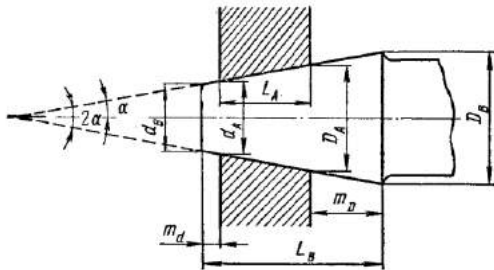


Рис. 1. Розрахункова схема конічного з'єднання

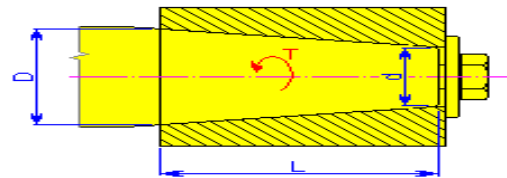


Рис. 2. Конічне з'єднання

Визначаємо розрахунок конусності:

$$C = 1 \div x = \frac{D-d}{L} = 2tg \frac{\alpha}{2}, \quad (1)$$

де d – менший діаметр валу, мм;

D – більший діаметр валу, мм;

L – довжина конуса, мм.

Розрахунок конічного з'єднання:

$$tg \frac{\alpha}{2} = \frac{D-d}{2L}. \quad (2)$$

Розрахунок середнього діаметра:

$$d_s = \frac{d+D}{2}, \text{ (мм)}. \quad (3)$$

Розрахунок необхідного тиску в з'єднанні:

$$p = \frac{2 \cdot 1000 \cdot k_s \cdot T \cdot \cos(\alpha/2)}{\pi \cdot d_s^2 \cdot L \cdot \vartheta}, \text{ (МПа)} \quad (4)$$

де p – тиск у з'єднанні, МПа;

k_s – запас міцності під час переміщення;

T – крутний момент, Нм;

ϑ – коефіцієнт затиску; ϑ – в межах з 0,008 до 0,1.

Розрахунок необхідної сили стиснення:

$$F_s = \pi \cdot d_s \cdot L \cdot p \cdot tg \left(\left(\frac{\alpha}{2} \right) + \varphi \right) = \pi \cdot d_s \cdot L \cdot p \cdot \frac{tg(\alpha/2) + \varphi}{1 - tg(\alpha/2) \cdot \varphi}, \text{ (N)} \quad (5)$$

де φ – кут тертя; $\varphi = \arctg \varphi$.

Розрахунок сили розриву з'єднання:

$$F_{s1} = \pi \cdot d_s \cdot L \cdot p \cdot tg \left(\varphi - \left(\frac{\alpha}{2} \right) \right) = \pi \cdot d_s \cdot L \cdot p \cdot \frac{\varphi - tg(\alpha/2)}{1 + \varphi \cdot tg(\alpha/2)}, \text{ (N)} \quad (6)$$

Розрахунок перевірки міцності:

$$p \leq p_A \quad (7)$$

де p_A – допустимий тиск, МПа.

T – крутний момент [Нм]



d – малий діаметр валу [мм]

D – більший діаметр валу [мм]

p – тиск у з'єднанні [МПа]

pA – допустимий тиск [МПа]

L – довжина втулки [мм]

U – затискний коефіцієнт $U =$ від 0,008 до 0,1 [-]

ks – запас;

ϕ – кут тертя; $\phi = \arctg \phi$.

Основна функція програмного забезпечення це введення вихідних даних та отримання результату для розрахунку конічного з'єднання. Програма мало функціональна, але завдання розрахунку виконує дуже добре [10-12].

Програмне забезпечення має відповідати наступним функціональним вимогам: правильність (функціонування відповідно до технічного завдання); точність (введення точних значень та отримання точних результатів, а не наближених до їх значення); сумісність (програмне забезпечення повинне підходити до різних видів операційних систем та прикладних програм (у які вона може вбудовуватися)); надійність (програма повинна бути захищена та має бути присутня система самоконтролю дій); універсальність (програма повинна працювати за будь-яких допустимих вихідних даних); захищеність (у програмному забезпеченні повинні бути засоби для захисту даних від зовнішнього впливу, які запобігають їх втраті в процесі розрахунків); корисність (розрахунки, що проводяться в системі, повинні бути корисними для виробничих робітників, і для студентів технічних ВУЗів); ефективність (програма має працювати з максимальною ефективністю для користувача); перевірюваність (значення та розрахунки, наведені в програмі повинні мати змогу перевірятися без будь-яких помилок або змін); адаптованість (програмне забезпечення повинне адаптовуватися під різні режими роботи та різних користувачів) [13-16].

Для розробки цієї програми використовувалося середовище розробки Microsoft Visual Studio 2022 [17].

Ця програма має такі мінімальні вимоги:

Операційна система: Microsoft Windows або macOS;

Мінімальна кількість оперативної пам'яті: 128 Мб;

Відеоадаптер: 32Мб, 64 Мб;

Процесор: 300 MHz та вище;

Жорсткий диск: 30 Мб;

Мова програмування: C# [18].

Основний матеріал дослідження

Процес створення програми (проекткування програми)

Для написання програми потрібно зробити наступні кроки:

1. Запустити Visual Studio 2010.
 2. У рядку меню вибрати *файл* → *створити* → *проект*.
 3. На панелі встановлених шаблонів розгорнути Visual C#, а потім вибрати *вікна*.
 4. У центральній області вибрати *консольний додаток* (або інший залежно від завдання у нашому випадку вибрати *створення форми*).
 5. У вікні *ім'я* вказуємо *ім'я проекту*, а потім натисніскаємо кнопку *дати*.
- У браузері з'явиться новий проект.
6. Якщо Program.cs не відкрито, в редакторі коду відкриваємо контекстне меню для Program.cs в браузері, а потім вибираємо *перейти до коду*. Відкривається вікно, де ми бачимо структуру проекту (рис. 3).

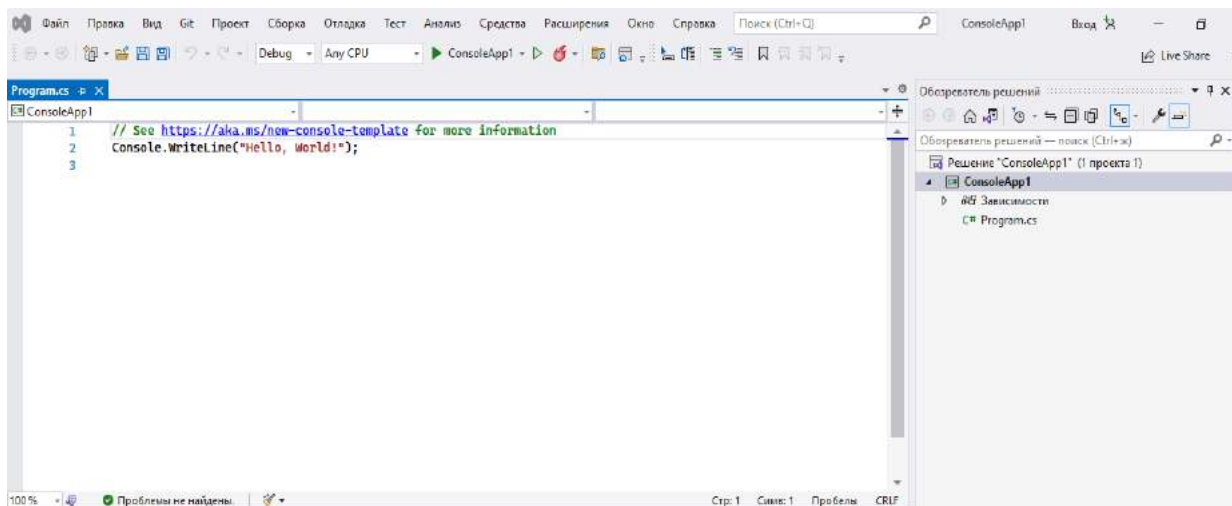


Рис. 3. Структура проекту



Проектування інтерфейсу

Інтерфейс користувача являє собою сукупність програмних та апаратних засобів, які забезпечують взаємодію між користувачем і комп'ютером. Діалоги становлять основу такої взаємодії. Під діалогом розуміється регламентований обмін інформацією між користувачем та комп'ютером, що здійснюється в часі й спрямований на спільне вирішення поставленої задачі.

Отже, для користувача найголовніше в кінцевий результат, а саме, легкість і зрозумілість інтерфейсу (рис. 4). Основна увага в роботі приділяється його проектуванню. На нашу думку, інтерфейс програми цілком підходить для середньостатистичного користувача.

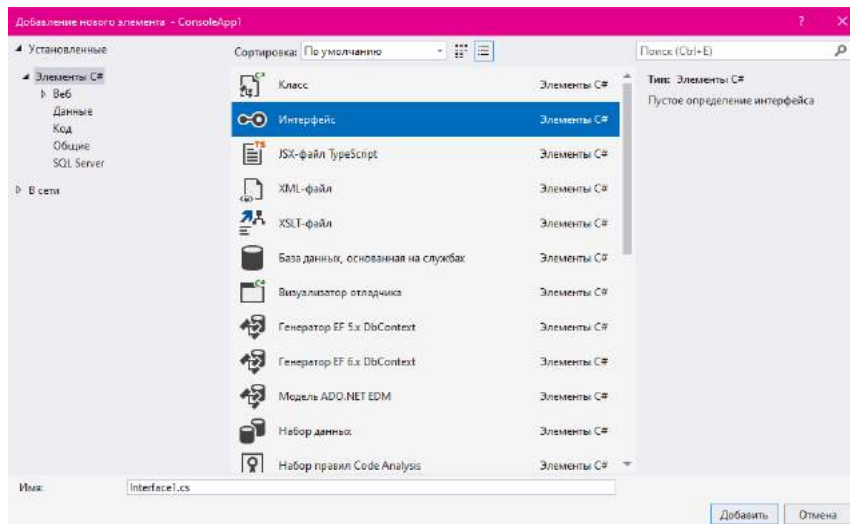


Рис. 3. Проектування інтерфейсу

Розробка програми

Перед початком розробки програми, потрібно запустити програму Visual Studio 2022:

Пуск → всі програми → Microsoft Visual Studio 2022.

Наступним кроком необхідно створити новий проект програми Windows Forms C# [19].

Для цього, натискаємо, створити новий проект, в настановних шаблонах вибираємо C# і вибираємо Windows Forms.

Повинно з'явитися вікно, як показано рисунку 4.

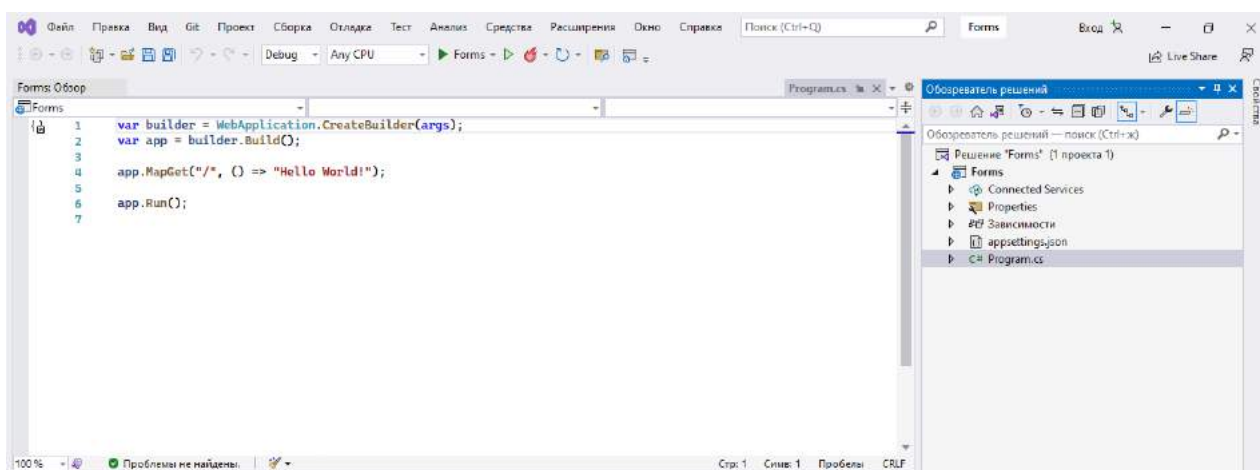


Рис. 4. Початкове вікно Windows Forms

Для того щоб наша форма набула коректного вигляду, для неї потрібно додати кнопки та інші функціональні речі (картинку, текстові бокси) Все це виконується з переходом на вкладку панель інструментів (рис. 5). Після цього в нашу форму додаємо різні функціональні елементи.

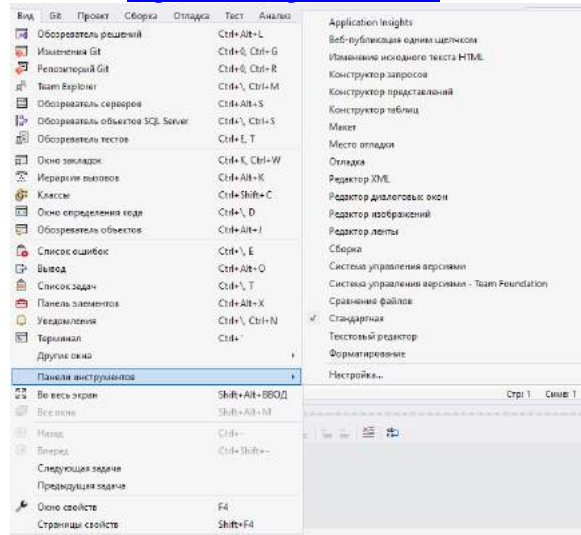


Рис. 5. Вкладка панели инструментов

Результаты досліджень

Для створення програмного забезпечення необхідно запрограмувати кожен елемент цієї форми. Спочатку визначимо тип вхідних даних у даному випадку це тип Double.

namespace Conic

```
{
public partial class Form1 : Form
{
string[] _size = {"[мм]", "[д]"};
string[] _force = {"[Н]", "[фТ сл]"};
string[] _moment = {"[Нм]", "[фН фТ]"};
string[] _pressure = {"[МПа]", "[фТ/кв.д]"};
public void calc(double D, double Dd, double L, double T, double pa, double v, double ks)
```

Також необхідно застерегти користувачів від неправильного введення даних. Це виконуємо в кінці програми.

```
catch
{
MessageBox.Show("Неправильно введені дані!");
}
}
```

Потрібно реалізувати розрахунок формул, це виконано так:

```
double C = (D - Dd) / L;
double tg = C / 2;
double ds = (D + Dd) / 2;
double cos = Math.Cos(Math.Atan(tg));
double p = (2000 * ks * T * cos) / (Math.PI * ds * ds * L * v);
double Fs = Math.PI * ds * L * p * (tg + v) / (1 - tg * v);
double Fs1 = Math.PI * ds * L * p * (v - tg) / (1 + tg * v);
tb_C.Text = Convert.ToString(C);
tb_tg.Text = Convert.ToString(tg);
tb_ds.Text = Convert.ToString(ds);
tb_p.Text = Convert.ToString(p);
tb_Fs.Text = Convert.ToString(Fs);
tb_Fs1.Text = Convert.ToString(Fs1);
```

Все гаразд, програма перемикає систему одиниць із метричної на британську (рис. 6).

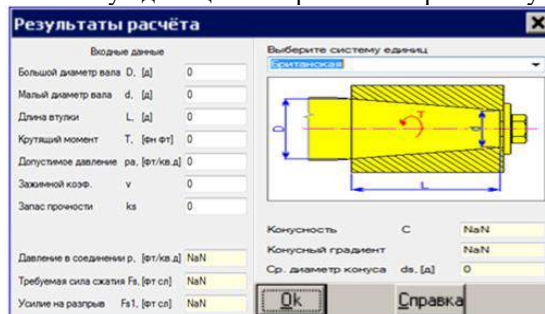


Рис. 6. Британська система одиниць



Проводимо перевірку на правильність введення (рис. 7).

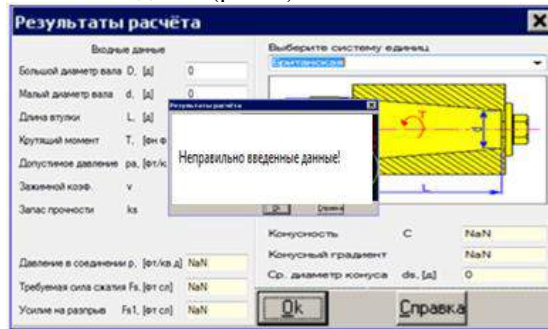


Рис. 7. Перевірка введення даних

Тестування показало, що програма не дозволяє вводити не коректні дані.

Отже, можна зробити один висновок, що програмне забезпечення функціонує правильно і повністю відповідає поставленим задачам.

Висновки

У науковій роботі написана програма додатку для розрахунку конічного з'єднання мовою програмування C# в середовищі програмування Visual Studio 2022. Програма була протестована, результати якої показали, що вона працює коректно та відповідає всім необхідним вимогам.

У науковій роботі були наведені переваги мови програмування C#. Мова C# дуже проста і легка для проектування, створення та використання програм. Її можна рекомендувати для початкового навчання програмування у вищих навчальних закладах, також, як альтернатива таким мовам програмування можуть бути Pascal та C++.

Список використаних джерел

- [1]. Farshidi, S., Jansen, S., Jong, R. & Brinkkemper, S. (2018). A decision support system for software technology selection. *Journal of Decision Systems*, 27 (1), P. 98–110. DOI: 10.1080/12460125.2018.1464821
- [2]. Becker, C., Kraхner, M., Plangg, M. & Rauber, A. (2013). Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria. 46th Hawaii international conference on system sciences (HICSS), P. 1193–1202. DOI: 10.1109/HICSS.2013.263
- [3]. Büyükköзkan, G., & Gülerüüz, S. (2016). A new integrated intuitionistic fuzzy group decision making approach for product development partner selection. *Computers & Industrial Engineering*, 102, 383–395. DOI: 10.1016/j.cie.2016.05.038
- [4]. Загора Я. П. Візуалізація процесу розрахунку механічних передач засобами APM TRANS Матеріали 5-ої Міжнародної науково-практичної конференції «Актуальні проблеми та перспективи технологічної і професійної освіти», 23-24 вересня 2016 року. Тернопіль: ТНПУ ім. В. Гнатюка, 2016. С. 38–40.
- [5]. Андреев І. А., Мікульонок І. О. Розрахунок, конструювання та надійність обладнання хімічних виробництв: термінологічний словник. Київ: ІВЦ "Видавництво «Політехніка»", 2002. 216 с.
- [6]. Franch, X., & Carvallo, J. P. (2003). Using quality models in software package selection. *IEEE Software*, 20 (1), 34–41. DOI: 10.1016/j.cie.2016.05.038
- [7]. Garg, R., Sharma, R., & Sharma, K. (2017). Mcdm based evaluation and ranking of commercial off-the-shelf using fuzzy based matrix method. *Decision Science Letters*, 6 (2), 117–136. DOI: 10.5267/j.dsl.2016.11.002
- [8]. Camps-Font, O., Rubianes-Porta, L., Valmaseda-Castellón, E., Jung, R. E., Gay-Escoda, C. & Figueiredo, R. (2021). Comparison of external, internal flat-to-flat, and conical implant abutment connections for implant-supported prostheses: A systematic review and network meta-analysis of randomized clinical trials. *The Journal of Prosthetic Dentistry*. 2021. 12 November, P. 1–14. DOI: 10.1016/j.prosdent.2021.09.029
- [9]. Jusoh, Y. Y., Chamili, K., Che Pa, N. & Yahaya, J. (2014). Open source software selection using an analytical hierarchy process (ahp). *American Journal of Software Engineering and Applications*, 3 (6), 83–89. DOI: 10.11648/j.ajsea.20140306.13
- [10]. Lin, H.-Y., Hsu, P.-Y., & Sheen, G.-J. (2007). A fuzzy-based decision-making procedure for data warehouse system selection. *Expert Systems with Applications*, 32 (3), 939–953. DOI: 10.1016/j.eswa.2006.01.031
- [11]. Montibeller, G., & Winterfeldt, D. (2015). Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, 35 (7), 1230–1251. DOI: 10.1111/risa.12360
- [12]. Ahmad, N. (2011). A Systematic Approach to Evaluating Open Source Software. *International Journal of Strategic Information Technology and Applications*. 2 (1): 48-67. DOI: 10.4018/jsita.2011010104.
- [13]. Onut, S., & Efendigil, T. (2010). A theoretical model design for erp software selection process under the constraints of cost and quality: A fuzzy approach. *Journal of Intelligent & Fuzzy Systems*, 21 (6), 365–378. DOI: 10.3233/IFS-2010-0457
- [14]. Ahmad N., (2011). A Systematic Approach to Evaluating Open Source Software. *International Journal of Strategic Information Technology and Applications*. 2 (1): 48-67. DOI: 10.4018/jsita.2011010104.



- [15]. Rodriguez, A., Ortega, F., & Concepción, R. (2017). An intuitionistic method for the selection of a risk management approach to information technology projects. *Information Sciences*, 375, 202–218. DOI: 10.1016/j.ins.2016.09.053
- [16]. Truong, A., Pham, C. (2021). Numerical modelling of power-actuated fastener connections joining high-strength steel sheet to mild steel plate subjected to monotonic shear. *Thin-Walled Structures*, 165. 107950–107962.
- [17]. Visual Studio 2022 version 17.0 Release Notes. docs.microsoft.com.
- [18]. The Roslyn .NET compiler provides C# and Visual Basic languages with rich code analysis APIs.: dotnet/roslyn. November 13, 2019. via GitHub.
- [19]. Welcome to C# 11. Retrieved December 10, 2022.

References

- [1]. Farshidi, S., Jansen, S., Jong, R. & Brinkkemper, S. (2018). A decision support system for software technology selection. *Journal of Decision Systems*, 27 (1), P. 98–110. DOI: 10.1080/12460125.2018.1464821
- [2]. Becker, C., Kraxner, M., Plangg, M. & Rauber, A. (2013). Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria. 46th Hawaii international conference on system sciences (HICSS), P. 1193–1202. DOI: 10.1109/HICSS.2013.263
- [3]. Büyükköçkan, G., & Gülleryüz, S. (2016). A new integrated intuitionistic fuzzy group decision making approach for product development partner selection. *Computers & Industrial Engineering*, 102, 383–395. DOI: 10.1016/j.cie.2016.05.038
- [4]. Zamora Y.P. Visualization of the process of calculating mechanical transmissions by means of APM TRANS Materials of the 5th International Scientific and Practical Conference "Actual Problems and Prospects of Technological and Vocational Education", September 23-24, 2016. Ternopil: TNPU named after V. Hnatyuk, 2016. P. 38–40.
- [5]. Andreev I. A., Mikulonok I. O. Calculation, design and reliability of chemical production equipment: terminology dictionary. Kyiv: Polytechnic Publishing House, 2002. 216 c.
- [6]. Franch, X., & Carvallo, J. P. (2003). Using quality models in software package selection. *IEEE Software*, 20 (1), 34–41. DOI: 10.1016/j.cie.2016.05.038
- [7]. Garg, R., Sharma, R., & Sharma, K. (2017). Mcdm based evaluation and ranking of commercial off-the-shelf using fuzzy based matrix method. *Decision Science Letters*, 6 (2), 117–136. DOI: 10.5267/j.dsl.2016.11.002
- [8]. Camps-Font, O., Rubianes-Porta, L., Valmaseda-Castellón, E., Jung, R. E., Gay-Escoda, C. & Figueiredo, R. (2021). Comparison of external, internal flat-to-flat, and conical implant abutment connections for implant-supported prostheses: A systematic review and network meta-analysis of randomized clinical trials. *The Journal of Prosthetic Dentistry*. 2021. 12 November, P. 1–14. DOI: 10.1016/j.prosdent.2021.09.029
- [9]. Jusoh, Y. Y., Chamili, K., Che Pa, N. & Yahaya, J. (2014). Open source software selection using an analytical hierarchy process (ahp). *American Journal of Software Engineering and Applications*, 3 (6), 83–89. DOI: 10.11648/j.ajsea.20140306.13
- [10]. Lin, H.-Y., Hsu, P.-Y., & Sheen, G.-J. (2007). A fuzzy-based decision-making procedure for data warehouse system selection. *Expert Systems with Applications*, 32 (3), 939–953. DOI: 10.1016/j.eswa.2006.01.031
- [11]. Montibeller, G., & Winterfeldt, D. (2015). Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, 35 (7), 1230–1251. DOI: 10.1111/risa.12360
- [12]. Ahmad, N. (2011). A Systematic Approach to Evaluating Open Source Software. *International Journal of Strategic Information Technology and Applications*. 2 (1): 48-67. DOI: 10.4018/jsita.2011010104.
- [13]. Onut, S., & Efendigil, T. (2010). A theoretical model design for erp software selection process under the constraints of cost and quality: A fuzzy approach. *Journal of Intelligent & Fuzzy Systems*, 21 (6), 365–378. DOI: 10.3233/IFS-2010-0457
- [14]. Ahmad N., (2011). A Systematic Approach to Evaluating Open Source Software. *International Journal of Strategic Information Technology and Applications*. 2 (1): 48-67. DOI: 10.4018/jsita.2011010104.
- [15]. Rodriguez, A., Ortega, F., & Concepción, R. (2017). An intuitionistic method for the selection of a risk management approach to information technology projects. *Information Sciences*, 375, 202–218. DOI: 10.1016/j.ins.2016.09.053
- [16]. Truong, A., Pham, C. (2021). Numerical modelling of power-actuated fastener connections joining high-strength steel sheet to mild steel plate subjected to monotonic shear. *Thin-Walled Structures*, 165. 107950–107962.
- [17]. Visual Studio 2022 version 17.0 Release Notes. docs.microsoft.com.
- [18]. The Roslyn .NET compiler provides C# and Visual Basic languages with rich code analysis APIs.: dotnet/roslyn. November 13, 2019. via GitHub.
- [19]. Welcome to C# 11. Retrieved December 10, 2022.

Отримана в редакції 10.04.2023. Прийнята до друку 08.05.2023. Received 10 April 2023. Approved 10 May 2023. Available in Internet 19 June 2023.