**КУРСОВИЙ ПРОЕКТ**

з дисципліни **« Програмування »**

на тему**: « Розробка програми калькулятор »**

Студента(ки) 3 курсу, групи 31Із

ПІБ Буртенко Костянтин Сергійович

Керівник к. т. н., ст. викладач Шаригін О.А.

(вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка

Кількість балів: оцінка ECTS

Члени комісії: к. п. н., доцент Габрусєв В.Ю.

ст. викладач Устименко О.Б.

к. т. н., ст. викладач Шаригін О.А.

Студент ПІБ

(підпис)

Зміст

[Вступ 3](#_Toc157032746)

[РОЗДІЛ 1. Огляд проблеми (задачі) та існуючі способи її вирішення 5](#_Toc157032747)

[1.1. Постановка задачі. 5](#_Toc157032748)

[1.2. Огляд проблеми, аналіз публікацій, обґрунтування актуальності вирішення проблеми. 5](#_Toc157032749)

[1.3. Огляд існуючих рішень. 6](#_Toc157032750)

[Висновки до І розділу 7](#_Toc157032751)

[РОЗДІЛ 2. Вирішення поставленої задачі 8](#_Toc157032752)

[2.1. Огляд та обґрунтування використовуваних технологій для розробки програмного засобу. Засобів розробки системи і тд. 8](#_Toc157032753)

[2.2. Алгоритм роботи розробленого програмного засобу. 8](#_Toc157032754)

[2.3. Інструкція використання розробленого програмного засобу (програмного засобу) 13](#_Toc157032755)

[Висновки для ІІ розділу 13](#_Toc157032756)

[Загальні Висновки 14](#_Toc157032757)

[Використанні джерела 15](#_Toc157032758)

[Додатки 16](#_Toc157032759)

# Вступ

Актуальність розробки програми калькулятор полягає в потребі зручного і простого додатку, який допоможе швидко і легко виконати прості математичні операції. Так як в наш час швидкість дуже цінується, розробка ефективного калькулятора стає актуальним завданням.

Важливість цієї розробки полягає в тому, що калькулятор — це не лише інструмент для виконання арифметичних операцій але і універсальний помічник у різних сферах. Його можна використовувати в повсякденному житті, навчанні або професійній діяльності. Калькулятор здатен забезпечити точність і швидкість обчислень, що робить його незамінним в нашому житті.

Існуючі підходи до розв’язання проблеми розробки програми калькулятор включають у себе розробку зрозумілого інтерфейсу та оптимізацію алгоритмів обчислень.

**Мета:** розробити програму калькулятор, яка допоможе користувачам розв’язувати базові приклади.

**Завдання дослідження:**

1. Проаналізувати літературу і існуючі реалізації.
2. Розробка механізму введення та перевірка правильності даних
3. Розробка застосунку Калькулятор
4. Протестувати розроблений застосунок

**Об’єктом дослідження** процес розробки застосунку калькулятор, який виконує прості операції з числами

**Предмет дослідження.** Підходи до розробки програми калькулятор з акцентом на оптимізацію обчислень і зручністю її у використанні.

**Методи дослідження:** аналіз матеріалів, які пов’язані з обраною темою. Аналіз і порівняння вже існуючих калькуляторів, які написанні мовою програмування Python, для отримання уявлення і розуміння яка будово і структура має бути.

**Особистий внесок автора.** Одним з основних моїх внесків в роботі, був зручний інтерфейс, який буде зрозумілий користувачу.Часто додатки роблять перенавантаженими, тоді користувач губиться і йому не зручно користуватись. Я намагався втілити навички UX, щоб досягти потрібного ефекту.

# РОЗДІЛ 1. Огляд проблеми (задачі) та існуючі способи її вирішення

## Постановка задачі.

В ході даної роботи передбачається визначення та аналіз задачі, пов'язаної із створенням калькулятора на мові програмування Python. Основний акцент було зроблено на формулюванні необхідних математичних операцій, які програма повинна вміти виконувати, а також на розробці ефективних алгоритмів для їх реалізації. Створення зрозумілого інтерфейсу, який буде неперенавантажений функціями і буде виконувати прості арифметичні дії.

## Огляд проблеми, аналіз публікацій, обґрунтування актуальності вирішення проблеми.

При автоматизації підрахунків існує ризик невірного визначення математичних операторів та їх правильного використання. В статті "Математичні (арифметичні) оператори. Приклади" [7] наведено чіткі приклади, які демонструють, як виконуються різні математичні дії.

Під час розробки програми я намагався створити додаток, який точно виконує прості математичні операції, такі як множення, ділення, додавання і віднімання. Крім того, в програмі буде реалізовано перевірку умов, яка сповіщає користувача у випадку виявлення помилок в введених даних.

## 1.3. Огляд існуючих рішень.

Я переглянув готові калькулятори на різних сайтах. Посилання на сайти в використаних джерелах.

В першому калькуляторі [5] я не бачу обробки помилок. Тобто програма не враховує можливість помилок в обчисленнях, такі як ділення на нуль, чи інші некоректні операції. Додавання обробки помилок поліпшить його функціональність і надійність. Мені сподобалось, що там є досить корисні функції, такі як відсотки і піднесення до квадрату, які я обов’язково додам в наступних оновленнях мого калькулятора.

В другому випадку [6] можу виділити як позитивні так і негативні сторони. З позитивних, код написано просто і зрозуміло, що дає можливість легко зрозуміти структуру і логіку. Також з позитивних сторін є графічний інтерфейс. З мінус це відсутність перевірки введених символів чи результату, що може призвести до некоректної роботи програми.

Якщо брати до уваги інші калькулятори то деякі були перенавантаженні різними функціями, які ми використовуємо не так часто. І через це користувачу може бути не зручно ним користуватись. Наступним рішенням було відсутність залежності від сторонніх розробників, я самостійно відповідаю за свою програму. За потреби я її вдосконалюю, виправляю недоліки.

Можливість використовувати той стиль, який мені подобається. Тобто, як мій калькулятор буде виглядати зовнішньо. Це теж важливий аспект, при виборі яким додатком користуватись.

## Висновки до І розділу

В ході розгляду проблеми, пов'язаної із створенням програми калькулятора на мові програмування Python, були визначені актуальні вимоги до розробки ефективного та зручного інструменту для математичних обчислень. Вивчення різних джерел дозволило здобути значущі уроки з попередніх досліджень та створити контекст для розуміння сучасних тенденцій у даній галузі.

Аналіз існуючих методів вирішення цієї проблеми підкреслив різноманіття підходів до розробки калькуляторів та використання мови програмування Python для цих програм.

Розробка програми калькулятора на мові програмування Python є актуальною завданням, і під час вдосконалення слід звертати увагу на оптимізацію алгоритмів, забезпечення зручності користувача та реалізацію сучасних функціональних можливостей.

# РОЗДІЛ 2. Вирішення поставленої задачі

## 2.1. Огляд та обґрунтування використовуваних технологій для розробки програмного засобу. Засобів розробки системи і тд.

1) Операційна система на якій було розроблено застосунок: Windows 11.

2) Бази даних не використовувались. Це обумовлене відсутністю необхідності зберіганняі обробки досить великих обсягів даних.

3) Мова програмування: Python. Вибір пав на цю мову, тому що вона зрозуміла, легка в використанні і за допомогою неї можна швидко розробляти нескладні додатки.

4) Інтерфейс: графічний, за допомогою бібліотеки Tkinter. Вибір пав на цю бібліотеку тому, що вона вбудована у стандартний пакет Python. Інша причина полягає в тому, що вона підтримується на багатьох операційних системах. Ще одним плюсом є простий та легкий інтерфейс для створення графічних віконних програм.

## 2.2. Алгоритм роботи розробленого програмного засобу.

**1) Алгоритму роботи ПЗ.**

Зображення, що містить текст, схема, ряд, План

Автоматично згенерований опис

**2) опис використаних стандартних функцій.**

* **Tk():** Створення вікна.
* **eval():** Використовується для виконання математичних виразів, які введені користувачем.
* **Button():** Створення кнопок.
* **Entry():** Створення введеного поля для вводу тексту.

**3) опис використаних функцій сторонніх бібліотек.**

* **Frame():** Створення контейнера для розміщення віджетів.

**4) опис глобальних констант, та змінних якщо є, та їх призначення**

* **expression:** Зберігає поточний введений вираз.
* **input\_field:** Введене поле, де користувач вводить вираз.

**5) опис власних функцій, об’єктів, методів та їх призначення.**

* **btn\_click(item):** Обробляє натискання кнопок та виконує відповідні дії.
* **bt\_clear():** Очищає введений вираз та введене поле.

**6) навести основні (ключові) фрагменти коду формування.**

*Інтерфейс GUI*

root = Tk()

root.geometry("268x288")

root.title("Калькулятор")

root.resizable(0, 0)

frame\_input = Frame(root)

frame\_input.grid(row=0, column=0, columnspan=4, sticky="nsew")

input\_field = Entry(frame\_input, font='Arial 15 bold', width=24, state="readonly")

input\_field.pack(fill=BOTH)

buttons = (('7', '8', '9', '/', '4'),

('4', '5', '6', '\*', '4'),

('1', '2', '3', '-', '4'),

('0', '.', '=', '+', '4')

)

*Обробка натискання кнопок та виклик функції btn\_click.*

def btn\_click(item):

global expression

try:

input\_field['state'] = "normal"

if expression and expression[-1] == '=':

expression = expression[:-1]

expression += item

input\_field.delete(0, END)

input\_field.insert(END, expression)

if item == '=':

result = str(eval(expression[:-1]))

input\_field.delete(0, END)

input\_field.insert(END, result)

expression = result + "="

input\_field['state'] = "readonly"

except (ZeroDivisionError, SyntaxError):

handle\_error('Помилка')

*Очищення введеного виразу та введеного поля на екрані калькулятора за допомогою функції bt\_clear*

ef bt\_clear():

global expression

expression = ""

input\_field['state'] = "normal"

input\_field.delete(0, END)

input\_field['state'] = "readonly"

В Додатку 1 буде розміщено код в повному обсязі, або перейти за посиланням на репозиторій.

## 2.3. Інструкція використання розробленого програмного засобу (програмного засобу)

Мій калькулятор має простий і зручний інтерфейс, який буде зрозумілий кожному.   
**Введення прикладів.** За допомогою кнопок з цифрами від 0 до 9 ми можемо вводити цифри. Кнопки з « / » « \* » « + » « - » дозволяють нам виконувати прості арифметичні дії. Кнопка «.» дозволяє нам вводити десяткові числа.

**Обчислення виразів.** Після введення виразу натискаємо « = » для обчислення. Результат має з’явитись на екрані. Потім ми можемо або продовжити обчислення з цією відповіддю або очистити поле натиснувши на « С ».

**Обробка помилок.**  Калькулятор обробляє деякі помилки. Наприклад: ділення на 0 або синтаксичні помилки. І виводить певні повідомлення на екран.

## Висновки для ІІ розділу

У другому розділі було розглянуто процес розробки, основні елементи коду, які функції використовуються і для чого. Також були наведені інструкції, як користуватись додатком калькулятор. Логіка роботи програми включає в себе введення простих математичних операцій та виразів і їх розв’язання. Програма має зручний інтерфейс і можливість завершення роботи за потреби користувача.

# Загальні Висновки

Під час розробки була створена програма калькулятор, яка розв’язує прості математичні операції, з графічним інтерфейсом. Програма гарно виконує свої функції, має зрозумілий інтерфейс, яким зручно буде користуватись. Звісно, є функції, які можна додати в майбутньому. Наприклад: Піднесення до квадрату, квадратний корінь.   
Також можна буде створити розширену версію, яка буде вже вираховувати складніші вирази. Але і існуючого функціоналу буде вистачати для щоденного користування.

Була наведена зручна інструкцію по користуванню додатком, в якій детально розписано кроки і можливості кожної кнопки. Рекомендації зрозуміють як дорослі так і учні навчальних закладів.

# Використанні джерела

1. Курс: Програмування, 3 СОІ та 2 ІСТ (22-23 н.р.).

URL: <https://moodle.fmif.udu.edu.ua/course/view.php?id=939>

1. Прості математичні операції

URL: <http://surl.li/plqsn>

1. Посилання на репозиторій.

URL: <https://github.com/bukoos/Coursework.Burtenko.git>

1. Путівник мовою програмування Python

URL: <https://pythonguide.rozh2sch.org.ua/?authuser=1>

1. Варіант готового калькулятора 1

URL: <https://habr.com/ru/articles/481074/>

6. Варіант готового калькулятора 2

URL: <https://habr.com/ru/articles/481074/>

7. Математичні оператори

URL: <https://www.bestprog.net/uk/2019/10/14/python-mathematical-arithmetic-operators-examples-ua/>

# Додатки

Додаток 1: Код програми калькулятор

from tkinter import \*

def btn\_click(item):

global expression

try:

input\_field['state'] = "normal"

if expression and expression[-1] == '=':

expression = expression[:-1]

expression += item

input\_field.delete(0, END)

input\_field.insert(END, expression)

if item == '=':

result = str(eval(expression[:-1]))

input\_field.delete(0, END)

input\_field.insert(END, result)

expression = result + "="

input\_field['state'] = "readonly"

except (ZeroDivisionError, SyntaxError):

handle\_error('Помилка')

def bt\_clear():

global expression

expression = ""

input\_field['state'] = "normal"

input\_field.delete(0, END)

input\_field['state'] = "readonly"

def handle\_error(error\_msg):

input\_field.delete(0, END)

input\_field.insert(0, error\_msg)

root = Tk()

root.geometry("268x288")

root.title("Калькулятор")

root.resizable(0, 0)

frame\_input = Frame(root)

frame\_input.grid(row=0, column=0, columnspan=4, sticky="nsew")

input\_field = Entry(frame\_input, font='Arial 15 bold', width=24, state="readonly")

input\_field.pack(fill=BOTH)

buttons = (('7', '8', '9', '/', '4'),

('4', '5', '6', '\*', '4'),

('1', '2', '3', '-', '4'),

('0', '.', '=', '+', '4')

)

expression = ""

button = Button(root, text='C', command=bt\_clear)

button.grid(row=1, column=3, sticky="nsew")

for row in range(4):

for col in range(4):

Button(root, width=2, height=3, text=buttons[row][col],

command=lambda r=row, c=col: btn\_click(buttons[r][c])).grid(row=row + 2, column=col, sticky="nsew", padx=1, pady=1)

root.mainloop()