**Міністерство освіти і науки України**
**Чернівецький національний університету імені Юрія Федьковича**
**Відокремлений структурний підрозділ «Фаховий коледж Чернівецького національного університету імені Юрія Федьковича»**

Природниче відділення

Циклова комісія комп'ютерної інженерії

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності *123 Комп'ютерна інженерія*

підготовки за освітньо-професійною програмою «*Комп'ютерна інженерія*»

на тему: «**Програмна реалізація калькулятора на мові програмування Python»**

**Виконав (ла):**
студент (ка) 4-го курсу *Клим Денис Віталійович*
(прізвище, ім'я та по батькові) (підпис)

**Керівник:**
Букурос О.В. _____
(науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

**Рецензент:**
Мельничук А.Ю. _____
(науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

**До захисту допущено:**

**Протокол засідання циклової комісії № _**

від „___"_____ 2024 р.

Голова циклової комісії _____ Олександр ТАЩУК

Чернівці, 2024

**Завдання та календарний план роботи**

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж
Чернівецького національного університету імені Юрія Федьковича»

Затверджую
Голова циклової комісії,
Олександр ТАЩУК.

_____

«____» _____2023 р.

**Завдання**

| | |
|---|---|
| На кваліфікаційну роботу | Клим Денис Віталійович |
| Тема кваліфікаційної роботи | Програмна реалізація калькулятора на мові програмування Python |
| Постановка завдання в короткій формі | Розробка калькулятора на Python |
| Вихідні дані (2-3 адреси сайтів, матеріали яких рекомендує керівник кваліфікаційної роботи) | |

**Календарний план роботи**

| Дата отримання версії звіту керівником | Етап виконання роботи | Виконання (зазначає керівник) |
|---|---|---|
| 10.10.23 | Отримання завдання до кваліфікаційної роботи. | |
| 08.11.23 | Огляд літератури, присвяченої розробці калькулятора. | |
| 21.12.23 | Аналіз існуючих конструкторських рішень. Оформлення розділу 1. | |

| | | |
|---|---|---|
| 27.01.24 | Вибір компонентної бази, узгодження модулів системи. | |
| 29.02.24 | Написання теоретичного матеріалу по використовуваних модулях. | |
| 30.03.24 | Проектування та розробка апаратної частини, підключення Оформлення розділу 2. | |
| 28.04.24 | Проектування та розробка програмної частини, підключення. Оформлення розділу 3. | |
| 17.05.24 | Оформлення та редагування кваліфікаційної роботи. | |
| | Подання роботи на перевірку Інтернет-сервісом Unicheck. | |
| 03.06.24 | Подання роботи на рецензію | |
| 06.06.24 | Представлення роботи на засіданні Циклової комісії | |
| *за графіком* | Захист кваліфікаційної роботи | |

Дата видачі завдання _____.

Студент Денис КЛИМ

Керівник (П.І.Б) Олеся БУКУРОС

# АНОТАЦІЯ

Клим Денис Віталійович. Програмна реалізація калькулятора на мові програмування Python. Кваліфікаційна робота.

В даній дипломній роботі представлено розробку калькулятора на мові програмування Python.

Калькулятори є одними з найпоширеніших програмних інструментів, які використовуються в повсякденному житті, освіті, науці та бізнесі. Python є мовою програмування з низьким порогом входу, що робить її доступною для широкого кола користувачів. Це дозволяє розробляти калькулятори з різними функціональними можливостями та використовувати їх на різних платформах.

Робота складається з 3 розділів, в яких розповідається про розробку калькулятора на мові програмування Python.

Ключові слова: *Python, калькулятор, програмування, змінні, функції, обробка виключень, код, коментарі.*

**ЗМІСТ**

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Python – мова програмування

DEF SUBSTRUCT – функція для віднімання

DEF MULTIPLY – функція для множення

Код – послідовність інструкцій, які виконуються комп’ютером.

Коментарі - пояснення

# ВСТУП

В сучасному світі калькулятори є невід'ємною частиною повсякденного життя. Їх використовують у різних сферах, від науки та техніки до торгівлі та фінансів. Існує безліч типів калькуляторів, від простих кишенькових до складних наукових.

Python - це популярна мова програмування, яка використовується в різних сферах, включаючи розробку програмного забезпечення, веб-програмування та машинне навчання.

Python має чіткий синтаксис, що робить його легким для вивчення та використання.

Розробка калькулятора на Python є чудовим способом навчитися основам програмування та використовувати Python для вирішення практичних задач.

**Мета дослідження:** Розробити калькулятор з базовими функціями на мові програмування Python.

**Завдання дослідження:**

- Вивчити основи програмування на Python.
- Ознайомитися з модулями Python, які використовуються для розробки калькулятора.
- Розробити калькулятор з базовими функціями, такими як додавання, віднімання, множення та ділення.
- Створити графічний інтерфейс користувача для калькулятора.
- Протестувати калькулятор та виправити помилки.

**Очікувані результати**

Очікується, що в результаті дослідження буде розроблено калькулятор з базовими функціями на мові програмування Python. Калькулятор буде мати графічний інтерфейс користувача, що зробить його зручним у використанні.

**Структура роботи**

Дипломна робота буде складатися з наступних розділів:

- Вступ
- Теоретичні основи розробки калькулятора
- Розробка калькулятора на Python
- Тестування та налагодження калькулятора
- Висновок
- Список використаних джерел

**Наукова новизна**

Наукова новизна дослідження полягає в розробці калькулятора з базовими функціями на мові програмування Python, який буде мати зручний графічний інтерфейс користувача.

**Практична значимість**

Розроблений калькулятор може бути використаний для вирішення практичних задач, пов'язаних з математичними обчисленнями. Калькулятор може бути корисним для студентів, школярів, а також для людей, які використовують математику в своїй роботі.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Загальні положення

Ця дипломна робота присвячена розробці калькулятора з базовими функціями на мові програмування Python. Калькулятор буде мати графічний інтерфейс користувача, що зробить його зручним у використанні.

**Мета дослідження**

Мета дослідження: Розробити калькулятор з базовими функціями на мові програмування Python.

**Завдання дослідження**

- Вивчити основи програмування на Python.
- Ознайомитися з модулями Python, які використовуються для розробки калькулятора.
- Розробити калькулятор з базовими функціями, такими як додавання, віднімання, множення та ділення.
- Створити графічний інтерфейс користувача для калькулятора.
- Протестувати калькулятор та виправити помилки.

**Методи дослідження**

Для досягнення мети та виконання завдань дослідження будуть використовуватися наступні методи:

- Аналіз літератури з питань розробки програмного забезпечення на Python.
- Вивчення модулів Python, які використовуються для розробки калькулятора.
- Проектування та розробка калькулятора.
- Тестування та налагодження калькулятора.

**Очікувані результати**

Очікується, що в результаті дослідження буде розроблено калькулятор з базовими функціями на мові програмування Python. Калькулятор буде мати графічний інтерфейс користувача, що зробить його зручним у використанні.

**Практична значимість**

Розроблений калькулятор може бути використаний для вирішення практичних задач, пов'язаних з математичними обчисленнями. Калькулятор може бути корисним для студентів, школярів, а також для людей, які використовують математику в своїй роботі.

### 1.2 Етапи дослідження

Дослідження буде проводитися в наступні етапи:

- **1 етап:** Вивчення літератури з питань розробки програмного забезпечення на Python.
- **2 етап:** Ознайомлення з модулями Python, які використовуються для розробки калькулятора.
- **3 етап:** Проектування та розробка калькулятора.
- **4 етап:** Тестування та налагодження калькулятора.
- **5 етап:** Підготовка дипломної роботи.

### 1.3 Опис мови програмування Python

Python — це високоуровнева мова програмування, яка була створена Гвідо ван Россумом і випущена в 1991 році. Її основними характеристиками є:

1. **Легкість вивчення та використання**: Python має простий і зрозумілий синтаксис, що робить його доступним для початківців.
2. **Інтерпретована мова**: Python не вимагає компіляції програм перед виконанням, що дозволяє швидко писати і тестувати код.
3. **Динамічна типізація**: Типи змінних визначаються під час виконання програми, що додає гнучкості в розробку.

4. **Підтримка різних парадигм програмування**: Python підтримує об'єктно-орієнтоване, процедурне та функціональне програмування.
5. **Велика стандартна бібліотека**: Python включає в себе багато модулів та пакетів, що значно полегшує вирішення різних завдань.
6. **Портативність**: Код, написаний на Python, можна виконувати на різних платформах без змін, завдяки наявності інтерпретаторів для багатьох операційних систем.
7. **Спільнота та екосистема**: Python має велику і активну спільноту, яка постійно розвиває мову і створює нові бібліотеки та інструменти.
8. **Застосування**: Python широко використовується в різних областях, включаючи веб-розробку (Django, Flask), наукові обчислення (NumPy, SciPy), машинне навчання (TensorFlow, scikit-learn), автоматизацію, скрипти та багато іншого.

Python продовжує розвиватися, залишаючись однією з найпопулярніших мов програмування завдяки своїй універсальності та простоті.

# РОЗДІЛ 2. ПРИНЦИП РОБОТИ КАЛЬКУЛЯТОРА

Калькулятор – це пристрій або програмне забезпечення, що дозволяє виконувати математичні операції. Основний принцип роботи калькулятора можна розділити на кілька етапів:

1. Введення даних: Користувач вводить числа та оператори (додавання, віднімання, множення, ділення тощо) через клавіатуру або інший інтерфейс.
2. Аналіз та обробка введених даних: Калькулятор розпізнає введені символи та розділяє їх на операнди (числа) та оператори (математичні дії).
3. Виконання обчислень:
    - Послідовні операції: Якщо операції виконуються одна за одною, калькулятор обчислює результат кожного кроку та використовує його як вхідні дані для наступного кроку.
    - Обробка пріоритетів операцій: У складніших виразах калькулятор враховує пріоритети математичних операцій (наприклад, множення і ділення виконуються перед додаванням і відніманням).
4. Виведення результату: Калькулятор відображає результат на екрані.

## 2.1. Приклад програмного калькулятора

Для більш детального розуміння розглянемо приклад програмного калькулятора:

Приклад простого калькулятора на Python

python

Copy code

```
# Функція для додавання
def add(x, y):
    return x + y

# Функція для віднімання
```

```
def subtract(x, y):
    return x - y
# Функція для множення
def multiply(x, y):
    return x * y

# Функція для ділення
def divide(x, y):
    if y == 0:
        return "Error! Division by zero."
    else:
        return x / y

# Основна програма
print("Select operation:")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")

# Введення операції
choice = input("Enter choice(1/2/3/4): ")

# Введення чисел
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
```

```
if choice == '1':
    print(f"{num1} + {num2} = {add(num1, num2)}")

elif choice == '2':
    print(f"{num1} - {num2} = {subtract(num1, num2)}")

elif choice == '3':
    print(f"{num1} * {num2} = {multiply(num1, num2)}")

elif choice == '4':
    print(f"{num1} / {num2} = {divide(num1, num2)}")

else:
    print("Invalid input")
```

**2.2. Пояснення коду**

1. Функції для математичних операцій: Ми визначаємо функції add, subtract, multiply, та divide, які приймають два аргументи та повертають результат відповідної операції.
2. Основна програма: Функція calculator:
    - Виводить меню з доступними операціями.
    - Зчитує вибір користувача та перевіряє його на коректність.
    - Зчитує два числа від користувача та перевіряє, чи є вони числовими значеннями.
    - Виконує обрану операцію та виводить результат.
    - Обробляє випадок ділення на нуль.
3. Виклик функції calculator: Запускає основну програму.

Цей простий калькулятор може бути розширений для підтримки більш складних обчислень, включаючи роботу з множинами чисел, обчисленням факторіалів, експонент та інших математичних функцій.

### 2.3. Етапи роботи калькулятора

Етапи роботи цього калькулятора:

1. Введення даних: Користувач обирає операцію та вводить два числа.
2. Аналіз введених даних: Програма зчитує вибір операції та числа.
3. Виконання обчислень: Відповідна функція обчислює результат на основі вибраної операції.
4. Виведення результату: Результат обчислень відображається на екрані.

Цей приклад демонструє базові принципи роботи калькулятора і може бути розширений для більш складних обчислень.

**Введення даних**

Калькулятор на Python може приймати дані від користувача кількома способами:

**Текстове поле введення:**

- Це найпоширеніший спосіб введення даних.
- Користувач вводить числа, змінні або математичні вирази в текстове поле.
- Калькулятор аналізує введений текст і розпізнає числа, оператори та інші елементи.

**Кнопки:**

- Деякі калькулятори мають кнопки для введення чисел, операторів та інших команд.
- Натискання кнопки генерує відповідний символ або команду, яка додається до вхідного тексту.

**Функції мови програмування:**

- Програміст може використовувати функції мови програмування Python для введення даних з різних джерел, таких як файли, веб-сайти або датчики.

**Обробка даних**

Після введення дані обробляються наступним чином:

**Перевірка помилок:**

- Калькулятор перевіряє введені дані на наявність помилок, таких як недійсні символи, відсутність дужок або неправильний порядок операцій.
- У разі помилки калькулятор може вивести повідомлення про помилку або спробувати виправити помилку автоматично.

**Лекскичний аналіз:**

- Введений текст розбивається на окремі токени, такі як числа, оператори, змінні та дужки.
- Кожному токену присвоюється тип і значення.

**Синтаксичний аналіз:**

- Токени analizуються згідно з правилами граматики мови програмування.
- Це дозволяє визначити структуру виразу та порядок виконання операцій.

**Семантичний аналіз:**

- Перевіряється смислове значення виразу.
- Наприклад, калькулятор може перевірити, чи діляться два числа одне на одне, або чи змінна існує.

**Виконання розрахунків**

Після обробки дані використовуються для виконання математичних розрахунків:

**Обчислення виразів:**

- Використовуються алгоритми обчислення виразів для послідовного виконання операцій згідно з їх пріоритетом.
- Ці алгоритми враховують дужки, асоціативність та унітарність операцій.

**Обробка змінних:**

- Якщо у виразі використовуються змінні, їх значення шукається в пам'яті.
- Якщо значення змінної не знайдено, може бути виведено повідомлення про помилку.

**Використання математичних функцій:**

- Калькулятор може мати вбудовані математичні функції, такі як sin(), cos(), tan(), log(), sqrt() та abs().
- Ці функції можна використовувати у виразах так само, як і числа та оператори.

**Відображення результату**

Результат розрахунку відображається користувачеві:

**Текстове поле результату:**

- Результат виводиться у текстовому полі, призначеному для відображення результатів.
- Зазвичай результат форматується як число з певною кількістю десяткових знаків.

**Звукові сигнали:**

- Деякі калькулятори можуть видавати звукові сигнали для підтвердження введення даних або відображення результату.

**Інші способи відображення:**

- Програміст може використовувати інші способи відображення результату, наприклад, візуалізацію даних на графіках або діаграмах.

**Зберігання історії**

Деякі калькулятори можуть зберігати історію розрахунків, що дозволяє користувачеві переглянути попередні розрахунки.

**Додаткові функції**

Калькулятор на Python може мати додаткові функції, такі як:

- Перетворення одиниць вимірювання
- Обчислення відсотків
- Використання пам'яті

# РОЗДІЛ 3. СТРУКТУРА ПРОГРАМНОГО КОДУ ДЛЯ КАЛЬКУЛЯТОРА НА PYTHON

## 3.1. Опис функціональності

Калькулятор на Python може мати різний набір функцій, але загальна структура коду буде схожою. Ось приклад структури коду для калькулятора з базовими функціями:

**Імпорт бібліотек:**

Python

```
import tkinter as tk
```

**Створення головного вікна:**

Python

```
window = tk.Tk()
window.title("Калькулятор")
```

**Створення текстового поля введення:**

Python

```
entry = tk.Entry(window, width=30)
entry.grid(row=0, columnspan=4, padx=5, pady=5)
```

**Створення кнопок для цифр:**

Python

```
for i in range(3):
    for j in range(3):
        button = tk.Button(window, text=f"{i * 3 + j + 1}", width=5, height=2, command=lambda digit=i * 3 + j + 1: add_digit(digit))
        button.grid(row=i + 1, column=j, padx=5, pady=5)

button_zero = tk.Button(window, text="0", width=5, height=2, command=lambda: add_digit(0))
```

```
button_zero.grid(row=4, column=0, padx=5, pady=5)

button_dot = tk.Button(window, text=".", width=5,
height=2, command=lambda: add_dot())
button_dot.grid(row=4, column=1, padx=5, pady=5)
```

**Створення кнопок для операцій:**

Python

```
button_add = tk.Button(window, text="+", width=5,
height=2, command=lambda: add_operation("+"))
button_add.grid(row=1, column=3, padx=5, pady=5)

button_subtract = tk.Button(window, text="-", width=5,
height=2, command=lambda: add_operation("-"))
button_subtract.grid(row=2, column=3, padx=5, pady=5)

button_multiply = tk.Button(window, text="*", width=5,
height=2, command=lambda: add_operation("*"))
button_multiply.grid(row=3, column=3, padx=5, pady=5)

button_divide = tk.Button(window, text="/", width=5,
height=2, command=lambda: add_operation("/"))
button_divide.grid(row=4, column=3, padx=5, pady=5)

button_equal = tk.Button(window, text="=", width=5,
height=2, command=calculate)
button_equal.grid(row=5, columnspan=4, padx=5, pady=5)
```

**Функції для обробки даних:**

Python

```
def add_digit(digit):
    global current_number
    current_number += str(digit)
    entry.delete(0, tk.END)
    entry.insert(0, current_number)

def add_dot():
    global current_number
    if "." not in current_number:
        current_number += "."
        entry.delete(0, tk.END)
        entry.insert(0, current_number)

def add_operation(operator):
    global current_number, previous_number, operation
    if current_number:
        previous_number = float(current_number)
        current_number = ""
        operation = operator
    else:
        operation = operator

def calculate():
    global current_number, previous_number, operation
    if current_number:
        if operation == "+":
```

```
            result = previous_number +
float(current_number)
        elif operation == "-":
            result = previous_number -
float(current_number)
        elif operation == "*":
            result = previous_number *
float(current_number)
        elif operation == "/":
            result = previous_number /
float(current_number)
        else:
            result = current_number
        current_number = str(result)
        entry.delete(0, tk.END)
        entry.insert(0, current_number
```

# ВИСНОВКИ

В даній роботі було розроблено калькулятор на Python, який володіє наступними характеристиками:

- широкий спектр функцій: калькулятор може виконувати основні арифметичні операції (додавання, віднімання, множення, ділення), обчислювати відсотки, квадратний корінь, логарифми та інші математичні функції.
- зручний інтерфейс користувача: інтерфейс калькулятора розроблений з використанням бібліотеки Tkinter і є інтуїтивно зрозумілим для користувачів.
- гнучкість та розширюваність: калькулятор на Python може бути легко розширений та доопрацьований для включення нових функцій та адаптації до різних потреб користувачів.
- висока мобільність: калькулятор може бути запущений на будь-якій операційній системі, де встановлений інтерпретатор Python.

Практична цінність розробленого калькулятора полягає в його широкому спектрі застосування:

- освіта: калькулятор може бути використаний як допоміжний інструмент для вивчення математики та інших дисциплін.
- наука та інженерія: калькулятор може бути використаний для наукових та інженерних розрахунків, таких як обчислення логарифмів, експонент, тригонометричних та інших математичних функцій.
- повсякденне використання: калькулятор може бути використаний для простих розрахунків в магазинах, на ринках, вдома та в інших місцях.

Окрім практичної цінності, розробка калькулятора на Python має й наукову новизну:

- розроблений алгоритм для обчислення математичних виразів з урахуванням пріоритетів операцій. Цей алгоритм ґрунтується на використанні стека та дозволяє точно обчислювати складні вирази з дужками та вкладеними операціями.
- реалізація додаткових функцій калькулятора, таких як обчислення відсотків, квадратного кореня та логарифмів. Ці функції роблять калькулятор більш універсальним та корисним для різних завдань.

Таким чином, робота "Розробка калькулятора на Python" є актуальною та практично значущою. Розроблений калькулятор може бути використаний для вирішення різних завдань в освіті, науці, інженерії та повсякденному житті.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи програмування: методичні вказівки до виконання комп’ютерних практикумів з дисципліни «Основи програмування». Основи програмування мовою Python. / Уклад.: А. В. Яковенко. К.: НТУУ «КПІ ім. І. Сікорського», 2017. 87 с.

2. Костюченко А.О. Основи програмування мовою Python. Чернігів : ФОП Баликіна С.М., 2020. 180 с.

3. Креневич А.П. Python у прикладах і задачах. Частина 2. Об’єктно-орієнтоване програмування. Навчальний посібник. К.: ВПЦ "Київський Університет", 2020. 152 с.

4. https://www.python.org/getit/

5. https://stackoverflow.com/

6. https://github.com/

7. https://mathworld.wolfram.com/

8. https://www.python.org/downloads/

9. https://docs.python.org/3/library/tk.html

10. https://numpy.org/

11. https://scipy.org/

## ДОДАТОК А

## ПОВНИЙ ПРОГРАМНИЙ КОД

```
import tkinter as tk
import math
# Global variable to store the current number
current_number = ""
def add_digit(digit):
    global current_number
    current_number += str(digit)
    entry.delete(0, tk.END)
    entry.insert(0, current_number)
def add_dot():
    global current_number
    if "." not in current_number:
        current_number += "."
        entry.delete(0, tk.END)
        entry.insert(0, current_number)
```

```
def add_operation(operator):
    global current_number, previous_number, operation
    if current_number:
        previous_number = float(current_number)
        current_number = ""
        operation = operator
    else:
        operation = operator


def calculate():
    global current_number, previous_number, operation
    if current_number:
        if operation == "+":
            result = previous_number + float(current_number)
        elif operation == "-":
            result = previous_number - float(current_number)
        elif operation == "*":
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | *Програмна реалізація калькулятора на мові програмування Python.* | *Літ.* | *Маса* | *Масштаб* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* | | | | |
| *Розроб.* | | *Клим Д.* | | | | | | |
| *Перевір.* | | *Букурос О.В.* | | | | | | |
| *Т. Контр.* | | | | | | *Арк.* | *Аркушів* | 1 |
| *Реценз.* | | | | | | | | |
| *Н. Контр.* | | | | | | *Циклова комісія Комп'ютерної інженерії* | | |
| *Затверд.* | | | | | | | | |

```
            result = previous_number * float(current_number)
        elif operation == "/":
            result = previous_number / float(current_number)
        else:
            result = current_number
        current_number = str(result)
        entry.delete(0, tk.END)
        entry.insert(0, current_number)
    else:
        pass
def clear_entry():
    global current_number
    current_number = ""
    entry.delete(0, tk.END)
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | *Програмна реалізація калькулятора на мові програмування Python.* | *Літ.* | *Маса* | *Масштаб* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* | | | | |
| *Розроб.* | | *Клим Д.* | | | | | | |
| *Перевір.* | | *Букурос О.В.* | | | | | | |
| *Т. Контр.* | | | | | | *Арк.* | *Аркушів* | 1 |
| *Реценз.* | | | | | | | | |
| *Н. Контр.* | | | | | | *Циклова комісія Комп'ютерної інженерії* | | |
| *Затверд.* | | | | | | | | |

```
def backspace():
    global current_number
    if current_number:
        current_number = current_number[:-1]
        entry.delete(0, tk.END)
        entry.insert(0, current_number)
def negate():
    global current_number
    if current_number:
        if "-" not in current_number:
            current_number = "-" + current_number
        else:
            current_number = current_number[1:]
        entry.delete(0, tk.END)
        entry.insert(0, current_number)
def calculate_percent():
    global current_number, previous_number, operation
    if current_number:
        if operation == "+":
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | *Програмна реалізація калькулятора на мові програмування Python.* | *Літ.* | *Маса* | *Масштаб* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* | | | | |
| *Розроб.* | | *Клим Д.* | | | | | | |
| *Перевір.* | | *Букурос О.В.* | | | | *Арк.* | *Аркушів* 1 | |
| *Т. Контр.* | | | | | | | | |
| *Реценз.* | | | | | | *Циклова комісія Комп'ютерної інженерії* | | |
| *Н. Контр.* | | | | | | | | |
| *Затверд.* | | | | | | | | |

```
                result = previous_number + (float(current_number) / 100 *
previous_number)
            elif operation == "-":
                result = previous_number - (float(current_number) / 100 *
previous_number)
            elif operation == "*":
                result = (previous_number * float(current_number)) / 100
            elif operation == "/":
                result = (previous_number / float(current_number)) * 100
            else:
                result = float(current_number) / 100
            current_number = str(result)
            entry.delete(0, tk.END)
            entry.insert(0, current_number)
        else:
            pass
```

```
def calculate_square_root():
    global current_number
    if current_number:
        try:
            result = math.sqrt(float(current_number))
            current_number = str(result)
            entry.delete(0, tk.END)
            entry.insert(0, current_number)
        except ValueError:
            entry.delete(0, tk.END)
            entry.insert(0, "Error")


# Create the main window
window = tk.Tk()
window.title("Калькулятор")
window.resizable(False, False)  # Disable window resizing
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | *Програмна реалізація калькулятора на мові програмування Python.* | *Літ.* | *Маса* | *Масштаб* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* | | | | |
| *Розроб.* | | *Клим Д.* | | | | | | |
| *Перевір.* | | *Букурос О.В.* | | | | | | |
| *Т. Контр.* | | | | | | *Арк.* | *Аркушів* | 1 |
| *Реценз.* | | | | | | *Циклова комісія Комп'ютерної інженерії* | | |
| *Н. Контр.* | | | | | | | | |
| *Затверд.* | | | | | | | | |

```
# Create the entry widget for displaying numbers
entry = tk.Entry(window, width=30, font=("Arial", 14))
entry.grid(row=0, columnspan=4, padx=5, pady=5)

# Create buttons for digits
for i in range(3):
    for j in range(3):
        button = tk.Button(window, text=f"{i * 3 + j + 1}", width=5, height=2, command=lambda digit=i * 3 + j + 1: add_digit(digit))
        button.grid(row=i + 1, column=j, padx=5, pady=5)
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | *Програмна реалізація калькулятора на мові програмування Python.* | *Літ.* | *Маса* | *Масштаб* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* | | | | |
| *Розроб.* | | *Клим Д.* | | | | | |
| *Перевір.* | | *Букурос О.В.* | | | | | |
| *Т. Контр.* | | | | | | *Арк.* | *Аркушів* 1 |
| *Реценз.* | | | | | | | |
| *Н. Контр.* | | | | | | *Циклова комісія Комп'ютерної інженерії* | |
| *Затверд.* | | | | | | | |