also demonstrates a practical example of use. The report aims to support the educational process in the field of computer science and programming, and also provides recommendations for the effective use of the module to develop skills in data visualization and computer graphics.

Keywords: *Python, Turtle module, graphics, animation, programming.*

Науковий керівник: Тищенко С.І.,
канд. пед., наук, доцент, доцент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

УДК 004.04

РОЗРОБКА НАЙПРОСТІШОГО КАЛЬКУЛЯТОРА В PYTHON

Сиротенко Віктор Дмитрович,
здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *У цій науковій доповіді розглядається процес розробки найпростішого калькулятора з використанням мови програмування Python. Акцент робиться на поясненні кожного кроку в процесі створення програми, забезпеченні зрозумілого коду для початківців і представленні основних концепцій програмування, таких як змінні, функції та обробка виключень. Крім того, наводяться приклади коду з детальними коментарями, щоб полегшити розуміння. Ця доповідь може бути корисною як для новачків, які хочуть навчитися основам програмування на Python, так і для досвідчених розробників, які шукають чітке та зрозуміле пояснення створення простого калькулятора.*

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

Програмування – це захоплюючий і творчий процес, який дозволяє створювати різноманітні програмні рішення для вирішення повсякденних проблем. Одним з найпопулярніших завдань для початківців у програмуванні є створення калькулятора. Хоча це може здатися простим завданням, воно насправді дозволяє ознайомитися з багатьма основними концепціями програмування, такими як змінні, функції, обробка виключень та взаємодія з користувачем.

Розглянемо крок за кроком процес розробки найпростішого калькулятора на мові програмування Python. Ми почнемо з визначення вимог, а потім перейдемо до реалізації коду, забезпечуючи детальні пояснення на кожному етапі. Крім того, ми обговоримо деякі потенційні вдосконалення та розширення функціональності нашого калькулятора.

Перш ніж розпочати процес програмування, важливо чітко визначити вимоги до нашого калькулятора. У цьому прикладі ми будемо реалізовувати калькулятор, який може виконувати основні арифметичні операції: додавання, віднімання, множення та ділення. Програма повинна взаємодіяти з користувачем, отримуючи два числа та операцію, яку потрібно виконати, а потім виводити результат.

Тепер, коли ми визначили вимоги, ми можемо розпочати реалізацію коду нашого калькулятора на Python. Розглянемо кожен крок детально.

Отримання введених даних від користувача

```
num1 = float(input("Введіть перше число: "))
num2 = float(input("Введіть друге число: "))
operation = input("Введіть операцію (+, -, *, /): ")
```

У цьому коді ми використовуємо функцію input() для отримання вхідних даних від користувача. Перші два рядки отримують два числа, конвертуючи їх у тип float (числа з плаваючою точкою), щоб підтримувати десяткові значення. Третій рядок отримує операцію, яку потрібно виконати, у вигляді рядка.

Визначення функції для виконання операції

```
def perform_operation(num1, num2, operation):
    if operation == "+":
        result = num1 + num2
    elif operation == "-":
        result = num1 - num2
    elif operation == "*":
        result = num1 * num2
    elif operation == "/":
        if num2 != 0:
            result = num1 / num2
        else:
            print("Помилка: ділення на нуль!")
            return
    else:
        print("Невідома операція")
        return
    print(f"Результат: {num1} {operation} {num2} = {result}")
```

Ця функція perform_operation приймає два числа та операцію як аргументи. Вона використовує серію умовних конструкцій if/elif/else для визначення операції, яку потрібно виконати, і виконує відповідні обчислення. У випадку ділення на нуль функція виводить повідомлення про помилку і завершує свою роботу. Після виконання операції функція виводить результат у зручному для читання форматі за допомогою f-рядка.

Обробка виключень та виклик функції try:

```
        perform_operation(num1, num2, operation)
    except ValueError:
        print("Помилка: некоректне введення даних.")
```

Цей код обгортає виклик функції perform_operation в блок try/except для обробки можливих помилок. Якщо користувач вводить некоректні дані, наприклад, рядок замість числа, виникає виключення ValueError. У такому випадку програма виводить відповідне повідомлення про помилку.

Приклад роботи програми:

```
Введіть перше число: 10
Введіть друге число: 3
Введіть операцію (+, -, *, /): /
Результат: 10.0 / 3.0 = 3.3333333333333335
```

Ми розглянули процес розробки найпростішого калькулятора на Python. Ми визначили вимоги, реалізували код, забезпечивши детальні пояснення на кожному етапі, і продемонстрували приклад роботи програми.

Створення такого калькулятора є чудовою вправою для початківців у програмуванні, оскільки воно охоплює багато основних концепцій, таких як змінні, функції, обробка виключень та взаємодія з користувачем. Крім того, цей проект можна легко розширити, додавши нові функції, такі як обчислення математичних виразів, історія операцій або графічний інтерфейс користувача.

Загалом, розробка калькулятора на Python є відмінною відправною точкою для вивчення програмування і розуміння основних принципів створення програмного забезпечення.

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.

Abstract: *This scientific report examines the process of developing the simplest calculator using the Python programming language. Emphasis is placed on explaining each step in the program creation process, providing understandable code for beginners, and introducing basic programming concepts such as variables, functions, and exception handling. In addition, code examples are provided with detailed comments to facilitate understanding. This talk can be useful both for beginners who want to learn the basics of Python programming and for experienced developers who are looking for a clear and understandable explanation of how to create a simple calculator.*

Keywords: *Python, calculator, programming, variables, functions, exception handling, code, comments.*

Науковий керівник: Тищенко С.І.,
канд. пед., наук, доцент , доцент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

УДК 004.62

РОБОТА З БАЗАМИ ДАНИХ В PYTHON: SQL ТА NOSQL

Тимошенко Єлизавета Сергіївна,
здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *В дослідженні розглянуто бази даних SQL та NoSQL і робота з ними в Python. Наведено приклад використання модуля sqlite3, що дозволяє працювати з локальними SQL базами даних без зовнішніх залежностей. Розглянуто можливість використання бібліотеки pymongo для роботи з MongoDB в Python. Також проаналізовано особливості баз даних SQL та NoSQL і сфери їх використання.*

Ключові слова: *SQL, NoSQL, модуль sqlite3, бібліотека pymongo.*

SQL база даних (Structured Query Language) – це колекція даних, яка організована та зберігається за допомогою системи управління базами даних (СУБД), що підтримує мову SQL для маніпулювання цими даними. Вона складається з таблиць, де дані представлені у вигляді рядків (записів) та стовпців (атрибутів). Широко використовується в різних сферах, таких як бізнес, наука, освіта та інформаційні технології, завдяки її здатності зберігати та оброблювати великі обсяги даних та забезпечувати ефективний доступ до неї.

Python має бібліотеку SQLite, яка дозволяє працювати з локальними SQL базами даних без зовнішніх залежностей, через вбудований модуль sqlite3. Для підключення до віддалених SQL баз даних, таких як MySQL або PostgreSQL, можна використовувати сторонні бібліотеки, такі як mysql-connector-python та psycopg2. Також популярним вибором є бібліотека SQLAlchemy, яка пропонує ORM (Object Relational Mapping) для більш високорівневої роботи з базами даних.

Приклад використання модуля sqlite3: