



Instituto Politécnico Nacional

Escuela Superior de Computo



Unidad de académica:

Paradigmas de programación

Práctica de laboratorio 3:

“Implementación de la programación funcional”

Equipo:

- Elizalde Hernández Alan
- Reyes Ruíz Yoselyn Estefany
- Solares Velasco Arturo Misael
 - Solís Lugo Mayra
- Toral Hernández Leonardo Javier

Grupo: 3CV1

Profesor: García Floriano Andrés

Fecha:

19 marzo 2024

Reporte de Práctica: Implementación de una Calculadora con Programación Funcional

Introducción

En el marco del aprendizaje de la programación funcional, se ha desarrollado una calculadora en Python que ofrece una variedad de operaciones matemáticas. Este proyecto se ha centrado en aplicar los principios de la programación funcional, priorizando la creación de funciones puras para cada operación, evitando la dependencia excesiva de la función principal.

Objetivo

El objetivo principal de esta práctica fue desarrollar una calculadora funcional en Python, utilizando funciones puras para cada operación matemática. Se propuso la implementación de al menos 10 funciones matemáticas, así como la creación de un menú de opciones para seleccionar la operación deseada.

Implementación

Para lograr el objetivo propuesto, se ha seguido el siguiente enfoque:

Definición de Funciones: Se han creado funciones para cada operación matemática requerida, siguiendo los principios de las funciones puras. Estas funciones realizan cálculos sin modificar ningún estado fuera de su alcance, lo que garantiza una mayor modularidad y reutilización del código.

A continuación, se muestra un ejemplo de las funciones que implementa la calculadora.

```
# Definir las funciones para las operaciones aritméticas
def suma(a, b):
    return a + b
def resta(a, b):
    return a - b
def multiplicacion(a, b):
    return a * b
def division(a, b):
    if b == 0:
        return "Error: No se puede dividir por cero."
    else:
        return a / b
```

Ilustración 1 Funciones para las operaciones aritméticas.

```

# Funciones para otras operaciones
def raiz_cuadrada(a):
    if a < 0:
        return "Error: No se puede calcular la raíz cuadrada de un número negativo."
    else:
        return math.sqrt(a)
def raiz_n(a, n):
    if a < 0 and n % 2 == 0:
        return "Error: No se puede calcular la raíz par de un número negativo."
    else:
        return a ** (1/n)
def exponencial(a, b):
    return a ** b
def seno(a):
    return math.sin(a)
def coseno(a):
    return math.cos(a)
def tangente(a):
    return math.tan(a)
def arcoseno(a):
    return math.asin(a)
def arcocoseno(a):
    return math.acos(a)
def arcotangente(a):
    return math.atan(a)

```

Ilustración 2 Funciones para otras operaciones.

Menú de Opciones: Se ha implementado un menú de opciones que permite al usuario seleccionar la operación matemática que desea realizar. Esto se ha logrado utilizando un bucle while que solicita al usuario que ingrese el nombre de la operación deseada.

Validación de Entrada: Se ha incorporado una validación de entrada para manejar posibles errores durante la interacción con el usuario. Esto incluye la comprobación de la validez de la operación ingresada, así como la detección de errores al ingresar números.

Función Principal: La función principal (calculadora) actúa como punto de entrada del programa. Se encarga de llamar a las funciones correspondientes según la operación seleccionada por el usuario y mostrar el resultado.

A continuación, se muestra el código de la función principal que se ejecutará al final y que llevará a cabo todas las otras funciones que se declararon anteriormente.

```
def calculadora():
    "suma": suma,
    "resta": resta,
    "multiplicacion": multiplicacion,
    "division": division,
    "raiz_cuadrada": raiz_cuadrada,
    "raiz_n": raiz_n,
    "exponencial": exponencial,
    "seno": seno,
    "coseno": coseno,
    "tangente": tangente,
    "arcseno": arcseno,
    "arccoseno": arccoseno,
    "arctangente": arctangente
}

while True:
    try:
        operacion = input("\nIngrese la operación (suma, resta, multiplicacion, division, raiz_cuadrada, raiz_n, exponencial, seno, coseno, tangente, arcseno, arccoseno, arctangente): ")
        if operacion in operaciones:
            if operacion == "raiz_cuadrada":
                num = float(input("Ingrese el número para calcular la raíz cuadrada: "))
                resultado = operaciones[operacion](num)
            elif operacion == "raiz_n":
                num = float(input("Ingrese el número: "))
                n = int(input("Ingrese el índice de la raíz: "))
                resultado = operaciones[operacion](num, n)
            elif operacion in ["seno", "coseno", "tangente", "arcseno", "arccoseno", "arctangente"]:
                num = float(input("Ingrese el ángulo en radianes: "))
                resultado = operaciones[operacion](num)
            else:
                num1 = float(input("\nIngrese el primer número: "))
                num2 = float(input("Ingrese el segundo número: "))
                resultado = operaciones[operacion](num1, num2)
            print("El resultado es:", resultado)
        else:
            print("Operación no válida. Intente nuevamente.")
    except ValueError:
        print("Entrada inválida. Intente nuevamente.")
    continuar = input("\n¿Desea realizar otra operación? (s/n): ")
    if continuar.lower() != "s":
        break
```

Ilustración 3 Implementación de la función 'calculadora' (principal)

```
if __name__ == "__main__":
    # Llamar a la función principal
    calculadora()
```

Ilustración 4 Punto de entrada del programa con la función principal.

Funcionamiento del Programa

El programa desarrollado es una calculadora funcional en Python que ofrece una variedad de operaciones matemáticas. A continuación, se describe detalladamente cómo funciona:

Inicio del Programa: Al ejecutar el programa, se inicia la función principal `calculadora()`.

Menú de Opciones: Se muestra al usuario un menú de opciones que incluye una lista de operaciones matemáticas disponibles, tales como suma, resta, multiplicación, división, raíz cuadrada, raíz n -ésima, exponencial, seno, coseno, tangente, arcseno, arccoseno y arctangente.

Selección de Operación: El usuario selecciona una operación ingresando el nombre correspondiente en el menú de opciones. El programa valida la entrada del usuario para asegurarse de que la operación seleccionada sea válida.

Entrada de Datos: Dependiendo de la operación seleccionada, el programa solicita al usuario que ingrese los datos necesarios para realizar el cálculo. Por ejemplo, para operaciones binarias como suma, resta, multiplicación y división, se solicitan dos números. Para operaciones unarias como raíz cuadrada, se solicita un único número.

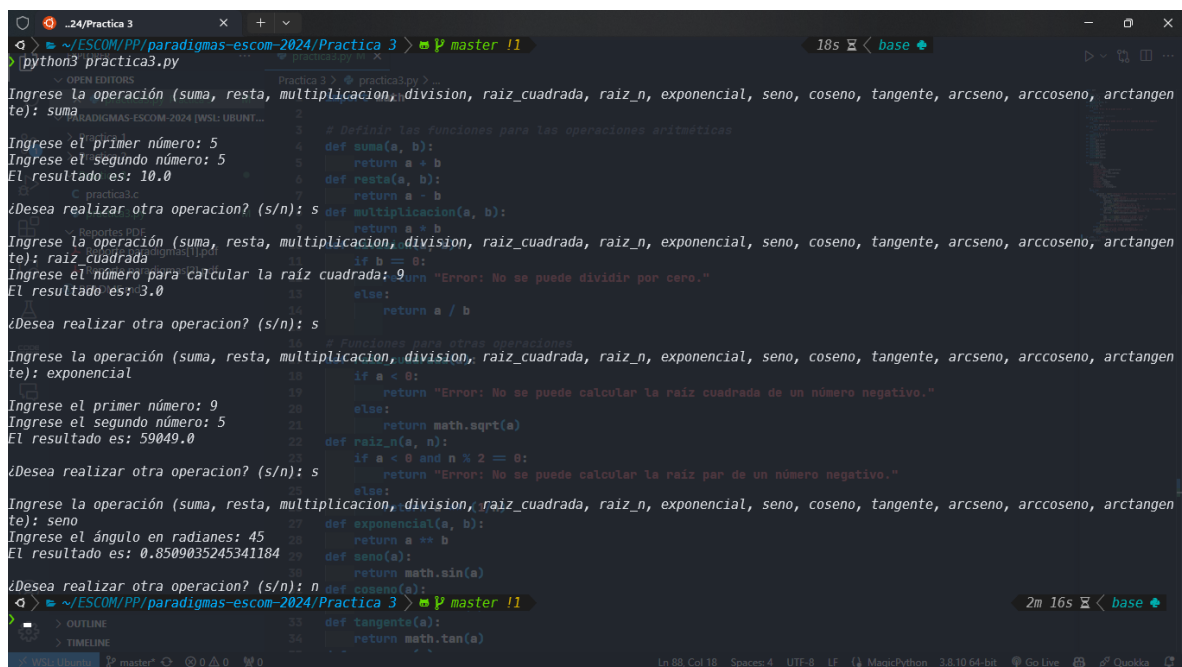
Cálculo y Presentación del Resultado: Una vez que se han ingresado los datos, el programa utiliza las funciones correspondientes para realizar el cálculo de la operación seleccionada. Estas funciones están diseñadas como funciones puras, lo que significa que no modifican ningún estado fuera de su ámbito local.

Manejo de Errores: El programa incluye mecanismos para manejar posibles errores durante la interacción con el usuario. Por ejemplo, se detectan y manejan errores cuando se ingresa una operación no válida o cuando se intenta realizar una división por cero.

Presentación del Resultado: Una vez completado el cálculo, el programa muestra el resultado al usuario en la misma consola donde se ejecuta el programa.

Repetición o Finalización: Después de mostrar el resultado, el programa pregunta al usuario si desea realizar otra operación. Si la respuesta es afirmativa, el programa vuelve al paso 2 y permite al usuario seleccionar una nueva operación. Si la respuesta es negativa, el programa finaliza su ejecución.

A continuación se muestra un ejemplo del programa en ejecución dentro de una consola Ubuntu-Linux.



```

24/Practica 3
~/ESCOM/PP/paradigmas-escom-2024/Practica 3 > master !1
python3 practica3.py
Ingrese la operación (suma, resta, multiplicacion, division, raiz_cuadrada, raiz_n, exponencial, seno, coseno, tangente, arcseno, arccoseno, arctangen
te): suma
Ingrese el primer número: 5
Ingrese el segundo número: 5
El resultado es: 10.0
¿Desea realizar otra operación? (s/n): s
Ingrese la operación (suma, resta, multiplicacion, division, raiz_cuadrada, raiz_n, exponencial, seno, coseno, tangente, arcseno, arccoseno, arctangen
te): raiz_cuadrada
Ingrese el número para calcular la raíz cuadrada: 9
El resultado es: 3.0
¿Desea realizar otra operación? (s/n): s
Ingrese la operación (suma, resta, multiplicacion, division, raiz_cuadrada, raiz_n, exponencial, seno, coseno, tangente, arcseno, arccoseno, arctangen
te): exponencial
Ingrese el primer número: 9
Ingrese el segundo número: 5
El resultado es: 59049.0
¿Desea realizar otra operación? (s/n): s
Ingrese la operación (suma, resta, multiplicacion, division, raiz_cuadrada, raiz_n, exponencial, seno, coseno, tangente, arcseno, arccoseno, arctangen
te): seno
Ingrese el ángulo en radianes: 45
El resultado es: 0.8509035245341184
¿Desea realizar otra operación? (s/n): n
~/ESCOM/PP/paradigmas-escom-2024/Practica 3 > master !1
```

Ilustración 5 Programa en ejecución dentro de la consola de comandos. Se puede observar el funcionamiento de diferentes operaciones y al final el cierre del programa.

Conclusión

La implementación de una calculadora funcional en Python ha permitido aplicar los principios de la programación funcional de manera efectiva. El uso de funciones puras ha mejorado la legibilidad, modularidad y mantenibilidad del código, mientras que el menú de opciones ofrece una experiencia de usuario intuitiva. Este proyecto ha demostrado cómo la programación funcional puede utilizarse para crear programas eficientes, estructurados, y ordenados.