Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 2

з курсу:

«Спеціалізовані мови програмування»

**Виконав:**

студент гр. ІТ-31

Роман ХРАПЧУН

**Прийняв:**

Сергій ЩЕРБАК

Львів 2023

**Тема:** Основи побудови об’єктно-орієнтованих додатків на Python

**Мета роботи:** Розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів

**Хід роботи**

***Завдання 1: Створення класу Calculator***

Створіть клас Calculator, який буде служити основою для додатка калькулятора.

***Завдання 2: Ініціалізація калькулятора***

Реалізуйте метод \_\_init\_\_ у класі Calculator для ініціалізації необхідних атрибутів або змінних.

***Завдання 3: Введення користувача***

Перемістіть функціональність введення користувача в метод у межах класу Calculator. Метод повинен приймати введення для двох чисел і оператора.

***Завдання 4: Перевірка оператора***

Реалізуйте метод у класі Calculator, щоб перевірити, чи введений оператор є дійсним (тобто одним із +, -, \*, /). Відобразіть повідомлення про помилку, якщо він не є дійсним.

***Завдання 5: Обчислення***

Створіть метод у класі Calculator, який виконує обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення).

***Завдання 6: Обробка помилок***

Реалізуйте обробку помилок у межах класу Calculator для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідні повідомлення про помилку.

***Завдання 7: Повторення обчислень***

Додайте метод до класу Calculator, щоб запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозвольте йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

***Завдання 8: Десяткові числа***

Модифікуйте клас Calculator для обробки десяткових чисел (плаваюча кома) для більш точних обчислень.

***Завдання 9: Додаткові операції***

Розширте клас Calculator, щоб підтримувати додаткові операції, такі як піднесення до степеня (^), квадратний корінь (√) та залишок від ділення (%).

***Завдання 10: Інтерфейс, зрозумілий для користувача***

Покращте інтерфейс користувача у межах класу Calculator, надавши чіткі запити, повідомлення та форматування виводу для зручності читання.

**Main.py:**

import math

class Calculator:

"""Клас, який представляє собою простий калькулятор."""

def \_\_init\_\_(self, language='ukrainian'):

"""Ініціалізація об'єкта калькулятора.

Args:

language (str): Обрана мова. За замовчуванням 'ukrainian'.

"""

self.result = None

self.history = []

self.language = language

def set\_language(self, language):

"""Встановлює обрану мову.

Args:

language (str): Обрана мова.

"""

self.language = language

def get\_language(self):

"""Повертає обрану мову.

Returns:

str: Обрана мова.

"""

return self.language

def get\_operator\_prompt(self):

"""Повертає текстовий запит для введення виразу користувачем.

Returns:

str: Текстовий запит.

"""

if self.language == 'english':

return "Enter an expression (e.g., '2 + 2' or 'exit'): "

elif self.language == 'ukrainian':

return "Введіть вираз (наприклад, '2 + 2' або 'вихід'): "

else:

return "Enter an expression (e.g., '2 + 2' or 'exit'): "

def get\_continue\_prompt(self):

"""Повертає текстовий запит для продовження виконання програми.

Returns:

str: Текстовий запит.

"""

if self.language == 'english':

return "Do you want to continue (yes/no)? "

elif self.language == 'ukrainian':

return "Бажаєте продовжити (так/ні)? "

else:

return "Do you want to continue (yes/no)? "

def calculate(self, num1, operator, num2):

"""Виконує обчислення на основі введених чисел та оператора.

Args:

num1 (float): Перше число.

operator (str): Оператор.

num2 (float): Друге число.

Returns:

float or str: Результат обчислення або повідомлення про помилку.

"""

try:

num1 = float(num1)

num2 = float(num2)

if operator == '+':

self.result = num1 + num2

elif operator == '-':

self.result = num1 - num2

elif operator == '\*':

self.result = num1 \* num2

elif operator == '/':

if num2 == 0:

return "Ділення на нуль" if self.language == 'ukrainian' else "Division by zero"

self.result = num1 / num2

elif operator == '^':

self.result = num1 \*\* num2

elif operator == '√':

if num1 < 0:

return "Від'ємне число" if self.language == 'ukrainian' else "Negative number"

self.result = math.sqrt(num1)

elif operator == '%':

if num2 == 0:

return "Ділення на нуль" if self.language == 'ukrainian' else "Division by zero"

self.result = num1 % num2

else:

return "Невірний оператор" if self.language == 'ukrainian' else "Invalid operator"

return self.result

except ValueError:

return "Не той формат числа" if self.language == 'ukrainian' else "Wrong number format"

def check\_operator(self, operator):

"""Перевіряє правильність оператора.

Args:

operator (str): Оператор.

Returns:

bool: True, якщо оператор правильний, інакше False.

"""

return operator in ('+', '-', '\*', '/')

def add\_to\_history(self, expression, result):

"""Додає вираз та результат в історію обчислень.

Args:

expression (str): Вираз.

result (float): Результат обчислення.

"""

self.history.append((expression, result))

def show\_history(self):

"""Виводить історію обчислень."""

if not self.history:

print("Історія порожня." if self.language == 'ukrainian' else "History is empty.")

else:

print("Історія обчислень:" if self.language == 'ukrainian' else "Calculation history:")

for idx, (expr, res) in enumerate(self.history, start=1):

print(f"{idx}. {expr} = {res}")

def user\_input(self):

"""Запитує вираз користувача та повертає його.

Returns:

str: Вираз користувача.

"""

expression = input(self.get\_operator\_prompt())

return expression

def perform\_calculation(self, expression):

"""Виконує обчислення та виводить результат.

Args:

expression (str): Вираз користувача.

"""

try:

num1, operator, num2 = map(str.strip, expression.split())

if self.check\_operator(operator):

result = self.calculate(num1, operator, num2)

if isinstance(result, str):

print(result)

else:

print(f"Результат: {result}")

self.add\_to\_history(expression, result)

else:

print("Невірний оператор" if self.language == 'ukrainian' else "Invalid operator")

except ValueError:

print("Невірний формат" if self.language == 'ukrainian' else "Invalid format")

def main\_loop(self):

"""Головний цикл програми."""

while True:

expression = self.user\_input()

if expression.lower() == 'вихід' or expression.lower() == 'exit':

break

self.perform\_calculation(expression)

self.show\_history()

next\_operation = input(self.get\_continue\_prompt())

if next\_operation.lower() != 'так' and next\_operation.lower() != 'yes':

break

print("Програма завершена." if self.language == 'ukrainian' else "Program terminated.")

def run\_lab2():

"""Функція для виконання програми."""

print("Виберіть мову (українська або англійська):")

selected\_language = input("Choose a language (ukrainian or english): ").lower()

while selected\_language not in ['ukrainian', 'english']:

print("Неправильно вибрана мова. Будь ласка, виберіть українську або англійську.")

print("Invalid language selection. Please choose ukrainian or english.")

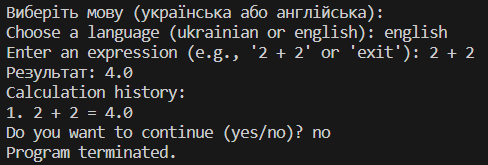
selected\_language = input("Choose a language (ukrainian or english): ").lower()

calculator = Calculator(selected\_language)

calculator.main\_loop()

if \_\_name\_\_ == "\_\_main\_\_":

run\_lab2()

**

*Рис.1 – Результат виконання програми*

**Висновок:** на даній лабораторній роботі я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.