МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

ОСНОВИ ПОБУДОВИ ОБ’ЄКТНО-ОРІЄНТОВАНИХ ДОДАТКІВ НА PYTHON

Виконала:

ст. гр. ІТ-21сп

Борисов П.Р.

Прийняв:

Щербак С.С.

Львів-2023

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

from BLL.lab2.CalculatorCore import CalculatorCore  
  
  
class CalculatorUI(CalculatorCore):  
  
 OPERATOR\_LIST = ['+', '-', '\*', '/', '%', '^', '√']  
  
 def get\_user\_choice(self):  
 return input("Enter your choice: ")  
  
 def get\_float\_input(self, prompt):  
 while True:  
 try:  
 return float(input(prompt))  
 except ValueError:  
 print("Incorrect number! Enter a valid number.")  
  
 def get\_operator(self):  
 while True:  
 operator = input('Enter your operator: ')  
 if operator in self.OPERATOR\_LIST:  
 return operator  
 else:  
 print('Incorrect operator.')  
  
 def perform\_calculation(self):  
 first\_number = self.get\_first\_number()  
 second\_number = self.get\_float\_input('Enter second number: ')  
 operator = self.get\_operator()  
  
 result = self.calculate\_number(first\_number, second\_number, operator)  
 self.display\_and\_store\_result(first\_number, second\_number, operator, result)  
  
 def get\_first\_number(self):  
 if self.memory and input('Get value from memory? Y/N: ').strip().upper() == 'Y':  
 return self.memory  
 return self.get\_float\_input('Enter first number: ')  
  
 def display\_and\_store\_result(self, first\_number, second\_number, operator, result):  
 expression = f"{first\_number} {operator} {second\_number}"  
 self.add\_to\_history(expression, result)  
 if self.settings["decimal\_places"] != 0:  
 result = round(result, self.settings["decimal\_places"])  
 print(result)  
 if self.settings["use\_memory"]:  
 self.store\_in\_memory(result)  
  
 def run\_calculator(self):  
 while True:  
 print("\nCalculator Menu:")  
 print("1. Perform calculation.")  
 print("2. View history.")  
 print("3. Change settings.")  
 print("4. Quit.")  
  
 user\_choice = self.get\_user\_choice()  
  
 if user\_choice == '1':  
 self.perform\_calculation()  
 elif user\_choice == '2':  
 print(self.view\_history())  
 elif user\_choice == '3':  
 self.change\_settings()  
 elif user\_choice == '4':  
 break

import math  
  
  
class CalculatorCore:  
 def \_\_init\_\_(self):  
 self.memory = None  
 self.history = []  
 self.settings = {  
 "decimal\_places": 2,  
 "use\_memory": True  
 }  
  
 def add\_to\_history(self, expression, value):  
 entry = f'{expression} - {value}'  
 self.history.append(entry)  
  
 def store\_in\_memory(self, value):  
 self.memory = value  
  
 def retrieve\_from\_memory(self):  
 return self.memory  
  
 def view\_history(self):  
 if not self.history:  
 return 'Empty history'  
 return '\n'.join(self.history)  
  
 def calculate\_number(self, first\_number, second\_number, operator):  
 try:  
 if operator == '-':  
 return first\_number - second\_number  
 if operator == '+':  
 return first\_number + second\_number  
 if operator == '\*':  
 return first\_number \* second\_number  
 if operator == '%':  
 return first\_number % second\_number  
 if operator == '^':  
 return first\_number \*\* second\_number  
 if operator == '√':  
 return math.sqrt(first\_number)  
 if operator == '/':  
 if second\_number == 0: # Prevent division by zero  
 raise ValueError("Error: Division by zero.")  
 return first\_number / second\_number  
 except Exception as e:  
 raise e  
  
 def change\_settings(self):  
 print("\nSettings Menu:")  
 print("1. Change decimal places.")  
 print("2. Toggle memory function.")  
 print("3. Exit to main menu.")  
  
 choice = input("Enter your choice: ")  
  
 if choice == '1':  
 self.change\_decimal\_places()  
 elif choice == '2':  
 self.toggle\_memory\_function()  
 elif choice == '3':  
 return  
 else:  
 print("Invalid choice. Please enter a number between 1 and 3.")  
  
 def change\_decimal\_places(self):  
 try:  
 places = int(input("Enter number of decimal places (0-10): "))  
 if 0 <= places <= 10:  
 self.settings["decimal\_places"] = places  
 print(f"Set decimal places to {places}.")  
 else:  
 print("Invalid number of decimal places.")  
 except ValueError:  
 print("Invalid input. Please enter a number between 0 and 10.")  
  
 def toggle\_memory\_function(self):  
 self.settings["use\_memory"] = not self.settings["use\_memory"]  
 status = "enabled" if self.settings["use\_memory"] else "disabled"  
 print(f"Memory function is now {status}.")

from BLL.lab2.CalculatorUI import CalculatorUI  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 ui = CalculatorUI()  
 ui.run\_calculator()

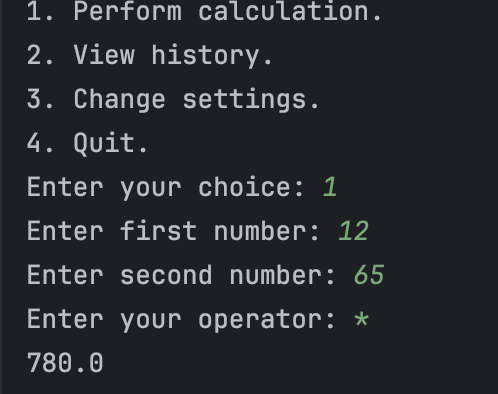


Рис 1. Виконання програми

**План роботи**

**Висновок:** я розробив консольний калькулятора в об’єктно орієнтованому стилі з використанням класів.