Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Людинно-машинна взаємодія*

Лабораторна робота № 9

Тема: Створення проекту «Калькулятор»

*Варіант 2*

Виконав: ***Гук Д.П.***

Група ІПЗ-31

Дата:2 грудня 2023 р.

Викладач: Пікуляк М.В.

Івано-Франківськ – 2023

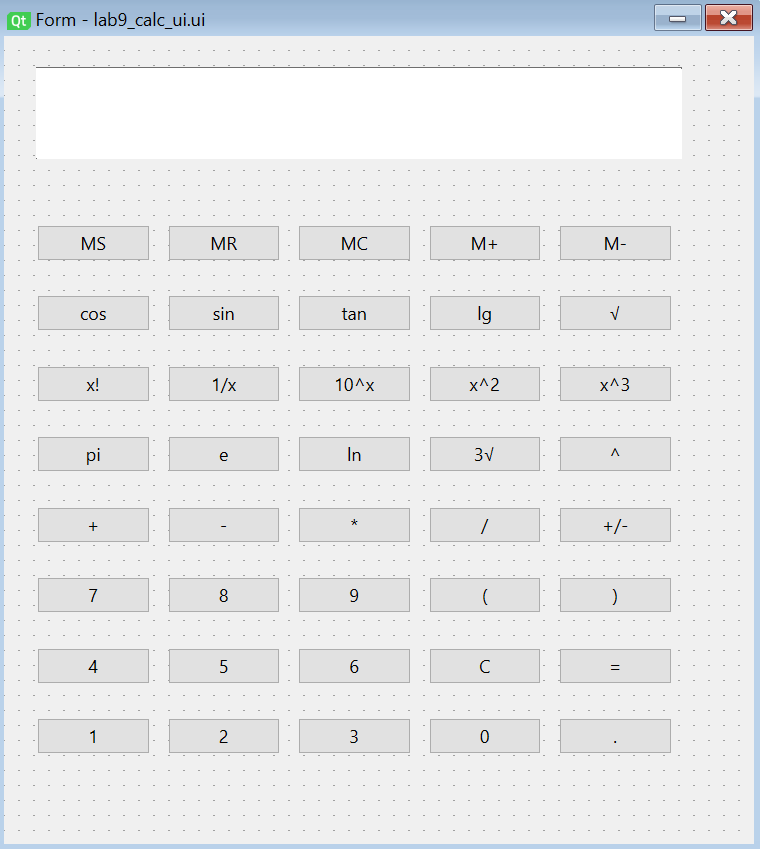
**Мета роботи:**

Створити проект «Калькулятор» за допомогою ICP “Qt-Creator”

**Завдання для виконання :**

Створити калькулятор за допомогою ICP “Qt-Creator”

**Текст скриптів і зображення діалогових вікон QtCreator з виконаним завданням :**

****

**Клас CalculatorLogic :**

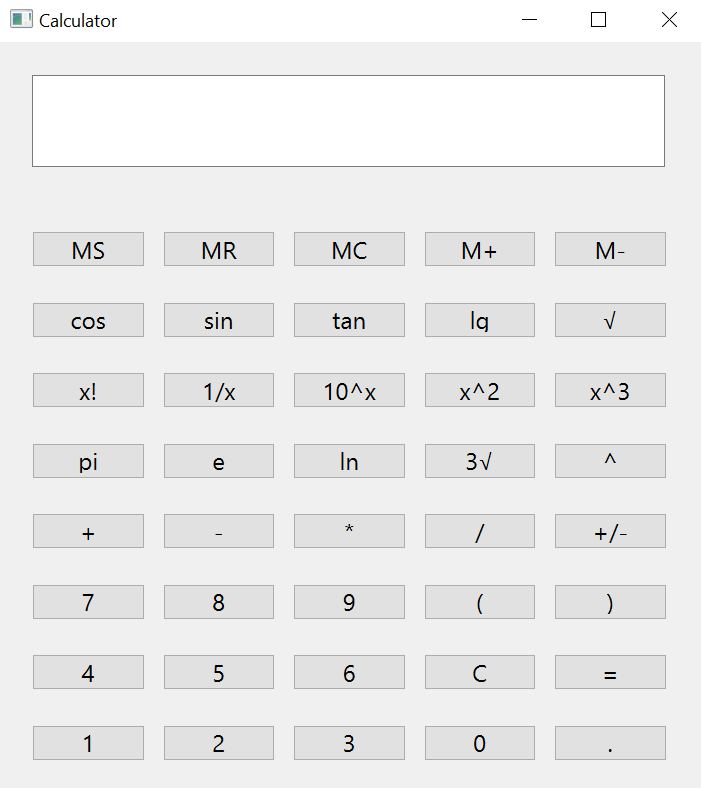
class CalculatorLogic:  
 def \_\_init\_\_(self, form):  
 self.form = form  
 self.setup\_buttons()  
 self.memory = []  
 self.pending\_operation = None  
 self.display\_text = ''  
 self.result\_shown = False  
  
  
 def setup\_buttons(self):  
 self.form.pushButton.clicked.connect(self.memory\_store)  
 self.form.pushButton\_17.clicked.connect(self.memory\_recall)  
 self.form.pushButton\_25.clicked.connect(self.memory\_clear)  
 self.form.pushButton\_31.clicked.connect(self.memory\_add)  
 self.form.pushButton\_26.clicked.connect(self.memory\_pop)  
  
 self.form.pushButton\_16.clicked.connect(self.addition)  
 self.form.pushButton\_22.clicked.connect(self.division)  
 self.form.pushButton\_19.clicked.connect(self.subtraction)  
 self.form.pushButton\_28.clicked.connect(self.multiplication)  
  
 self.form.pushButton\_2.clicked.connect(self.cos)  
 self.form.pushButton\_15.clicked.connect(self.sin)  
 self.form.pushButton\_21.clicked.connect(self.tan)  
  
 self.form.pushButton\_37.clicked.connect(self.factorial)  
 self.form.pushButton\_4.clicked.connect(self.power)  
 self.form.pushButton\_24.clicked.connect(self.square\_root)  
 self.form.pushButton\_32.clicked.connect(self.three\_root)  
 self.form.pushButton\_40.clicked.connect(self.power\_of\_ten)  
 self.form.pushButton\_38.clicked.connect(self.power\_two)  
 self.form.pushButton\_39.clicked.connect(self.power\_three)  
 self.form.pushButton\_36.clicked.connect(self.power\_minus\_one)  
 self.form.pushButton\_30.clicked.connect(self.logarithm\_10)  
 self.form.pushButton\_27.clicked.connect(self.logarithm\_nat)  
  
 self.form.pushButton\_5.clicked.connect(lambda: self.add\_to\_display('7'))  
 self.form.pushButton\_6.clicked.connect(lambda: self.add\_to\_display('4'))  
 self.form.pushButton\_7.clicked.connect(lambda: self.add\_to\_display('1'))  
 self.form.pushButton\_18.clicked.connect(lambda: self.add\_to\_display('8'))  
 self.form.pushButton\_11.clicked.connect(lambda: self.add\_to\_display('5'))  
 self.form.pushButton\_9.clicked.connect(lambda: self.add\_to\_display('2'))  
 self.form.pushButton\_23.clicked.connect(lambda: self.add\_to\_display('9'))  
 self.form.pushButton\_10.clicked.connect(lambda: self.add\_to\_display('6'))  
 self.form.pushButton\_34.clicked.connect(lambda: self.add\_to\_display('3'))  
 self.form.pushButton\_35.clicked.connect(lambda: self.add\_to\_display('0'))  
 self.form.pushButton\_3.clicked.connect(lambda: self.add\_to\_display('3.14'))  
 self.form.pushButton\_20.clicked.connect(lambda: self.add\_to\_display('2.72'))  
 self.form.pushButton\_13.clicked.connect(lambda: self.add\_to\_display('('))  
 self.form.pushButton\_41.clicked.connect(lambda: self.add\_to\_display(')'))  
 self.form.pushButton\_29.clicked.connect(self.change\_sign)  
 self.form.pushButton\_33.clicked.connect(self.decimal\_pressed)  
  
  
 self.form.pushButton\_8.clicked.connect(self.calculate\_result)  
 self.form.pushButton\_14.clicked.connect(self.clear\_display)  
  
  
  
 def change\_sign(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 value \*= -1  
 self.form.lineEdit.setText(str(value))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def decimal\_pressed(self):  
 current\_text = self.form.lineEdit.text()  
 if '.' not in current\_text:  
 self.form.lineEdit.setText(current\_text + '.')  
  
 def memory\_pop(self):  
 index = self.get\_memory\_index()  
 try:  
 if index is not None and 0 <= index < len(self.memory):  
 self.form.lineEdit.setText("")  
 self.memory\_index = index  
 self.form.lineEdit.setPlaceholderText("Enter value to subtract")  
 self.form.pushButton\_26.clicked.disconnect()  
 self.form.pushButton\_26.clicked.connect(lambda: self.memory\_store\_value('pop'))  
 else:  
 self.form.lineEdit.setText("Invalid memory index")  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def memory\_add(self):  
 index = self.get\_memory\_index()  
 try:  
 if index is not None and 0 <= index < len(self.memory):  
 self.form.lineEdit.setText("")  
 self.memory\_index = index  
 self.form.lineEdit.setPlaceholderText("Enter value to add")  
 self.form.pushButton\_31.clicked.disconnect()  
 self.form.pushButton\_31.clicked.connect(lambda: self.memory\_store\_value('add'))  
 else:  
 self.form.lineEdit.setText("Invalid memory index")  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def get\_memory\_index(self):  
 try:  
 index = int(self.form.lineEdit.text())  
 return index  
 except ValueError:  
 return None  
  
 def memory\_store\_value(self, operation):  
 try:  
 value = float(self.form.lineEdit.text())  
 if hasattr(self, 'memory\_index'):  
 index = self.memory\_index  
 if operation == 'add':  
 self.memory[index] += value  
 delattr(self, 'memory\_index')  
 self.form.lineEdit.setPlaceholderText("")  
 self.form.pushButton\_31.clicked.disconnect()  
 self.form.pushButton\_31.clicked.connect(self.memory\_add)  
 self.form.lineEdit.setText("")  
 elif operation == 'pop':  
 self.memory[index] -= value  
 delattr(self, 'memory\_index')  
 self.form.lineEdit.setPlaceholderText("")  
 self.form.pushButton\_26.clicked.disconnect()  
 self.form.pushButton\_26.clicked.connect(self.memory\_pop)  
 self.form.lineEdit.setText("")  
 else:  
 self.form.lineEdit.setText("Please select a memory index first")  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def memory\_store(self):  
 input\_text = self.form.lineEdit.text()  
 if input\_text:  
 try:  
 self.memory.append(float(input\_text))  
 self.form.lineEdit.setText("")  
 except ValueError:  
 self.form.lineEdit.setText("Invalid input for memory")  
 else:  
 self.form.lineEdit.setText("")  
  
 def memory\_recall(self):  
 try:  
 index = int(self.form.lineEdit.text())  
 if 0 <= index < len(self.memory):  
 recalled\_value = self.memory[index]  
 self.form.lineEdit.setText(str(recalled\_value))  
 elif len(self.memory) == 0:  
 self.form.lineEdit.setText("Memory is empty")  
 else:  
 self.form.lineEdit.setText("Index out of range")  
 except ValueError:  
 self.form.lineEdit.setText("Invalid index")  
  
 def memory\_clear(self):  
 self.memory.clear()  
  
  
  
 def cos(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.cos(math.radians(value))  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
  
  
 def sin(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.sin(math.radians(value))  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def tan(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.tan(math.radians(value))  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def logarithm\_10(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.log10(value)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def logarithm\_nat(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.log(value)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def factorial(self):  
 try:  
 value = int(self.form.lineEdit.text())  
 result = math.factorial(value)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
 except OverflowError:  
 self.form.lineEdit.setText("Result too large")  
  
 def addition(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '+' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' + ')  
 else:  
 parts = current\_text.split(' + ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 + num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
  
 def division(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '/' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' / ')  
 else:  
 parts = current\_text.split(' / ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 / num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
  
 def subtraction(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '-' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' - ')  
 else:  
 parts = current\_text.split(' - ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 - num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
  
 def modulus(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '%' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' % ')  
 else:  
 parts = current\_text.split(' % ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 % num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
  
 def multiplication(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '\*' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' \* ')  
 else:  
 parts = current\_text.split(' \* ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 \* num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
  
 def power(self):  
 try:  
 current\_text = self.form.lineEdit.text()  
 if '\*\*' not in current\_text:  
 self.form.lineEdit.setText(current\_text + ' \*\* ')  
 else:  
 parts = current\_text.split(' \*\* ')  
 if len(parts) == 2:  
 num1, num2 = map(float, parts)  
 result = num1 \*\* num2  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")  
 def square\_root(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.sqrt(value)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def three\_root(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.pow(value, (1/3))  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def power\_two(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.pow(value, 2)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def power\_minus\_one(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = 1 / value  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def power\_three(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.pow(value, 3)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def power\_of\_ten(self):  
 try:  
 value = float(self.form.lineEdit.text())  
 result = math.pow(10, value)  
 self.form.lineEdit.setText(str(result))  
 except ValueError:  
 self.form.lineEdit.setText("Error")  
  
 def add\_to\_display(self, value):  
 current\_text = self.form.lineEdit.text()  
 if self.result\_shown:  
 self.result\_shown = False  
 self.display\_text = ''  
 self.form.lineEdit.setText(current\_text + value)  
  
 def clear\_display(self):  
 self.form.lineEdit.setText("")  
 self.display\_text = ''  
 self.result\_shown = False  
  
 def calculate\_result(self):  
 try:  
 expression = self.form.lineEdit.text()  
 result = eval(expression)  
 self.form.lineEdit.setText(str(result))  
 self.result\_shown = True  
 except Exception as e:  
 self.form.lineEdit.setText("Error")

**Клас Ui\_Form :**

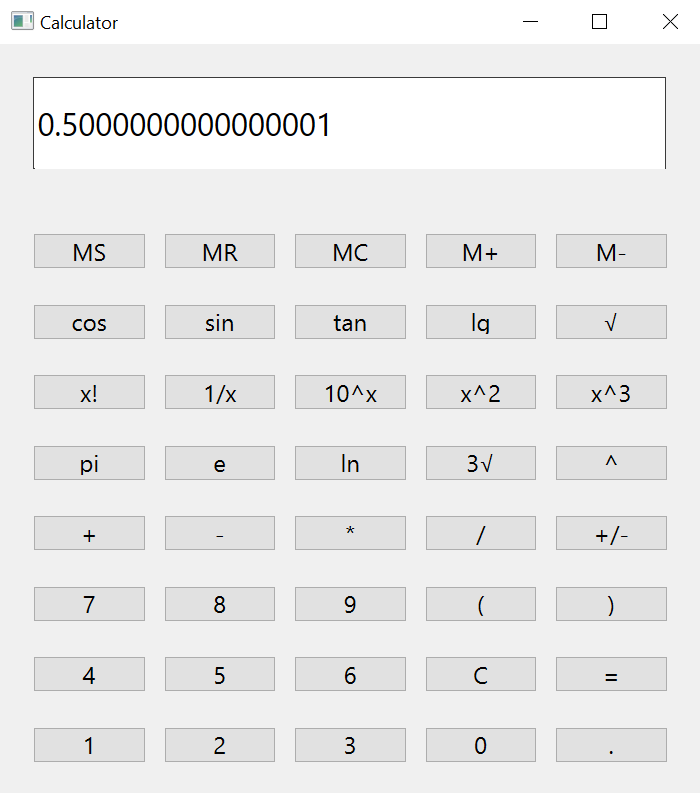
class Ui\_Form(object):  
 def setupUi(self, Form):  
 Form.setObjectName("Calculator")  
 Form.resize(470, 500)  
 self.lineEdit = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit.setGeometry(QtCore.QRect(22, 22, 422, 61))  
 self.lineEdit.setObjectName("lineEdit")  
 lineEdit\_font = self.lineEdit.font()  
 lineEdit\_font.setPointSize(16)  
 self.lineEdit.setFont(lineEdit\_font)  
 self.pushButton = QtWidgets.QPushButton(parent=Form)  
 self.pushButton.setGeometry(QtCore.QRect(22, 126, 75, 24))  
 self.pushButton.setObjectName("pushButton")  
 self.pushButton\_2 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_2.setGeometry(QtCore.QRect(22, 173, 75, 24))  
 self.pushButton\_2.setObjectName("pushButton\_2")  
 self.pushButton\_3 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_3.setGeometry(QtCore.QRect(22, 267, 75, 24))  
 self.pushButton\_3.setObjectName("pushButton\_3")  
 self.pushButton\_4 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_4.setGeometry(QtCore.QRect(370, 267, 75, 24))  
 self.pushButton\_4.setObjectName("pushButton\_4")  
 self.pushButton\_5 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_5.setGeometry(QtCore.QRect(22, 361, 75, 24))  
 self.pushButton\_5.setObjectName("pushButton\_5")  
 self.pushButton\_6 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_6.setGeometry(QtCore.QRect(22, 408, 75, 24))  
 self.pushButton\_6.setObjectName("pushButton\_6")  
 self.pushButton\_7 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_7.setGeometry(QtCore.QRect(22, 455, 75, 24))  
 self.pushButton\_7.setObjectName("pushButton\_7")  
 self.pushButton\_8 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_8.setGeometry(QtCore.QRect(370, 408, 75, 24))  
 self.pushButton\_8.setObjectName("pushButton\_8")  
 self.pushButton\_9 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_9.setGeometry(QtCore.QRect(109, 455, 75, 24))  
 self.pushButton\_9.setObjectName("pushButton\_9")  
 self.pushButton\_10 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_10.setGeometry(QtCore.QRect(196, 408, 75, 24))  
 self.pushButton\_10.setObjectName("pushButton\_10")  
 self.pushButton\_11 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_11.setGeometry(QtCore.QRect(109, 408, 75, 24))  
 self.pushButton\_11.setObjectName("pushButton\_11")  
 self.pushButton\_13 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_13.setGeometry(QtCore.QRect(283, 361, 75, 24))  
 self.pushButton\_13.setObjectName("pushButton\_13")  
 self.pushButton\_14 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_14.setGeometry(QtCore.QRect(283, 408, 75, 24))  
 self.pushButton\_14.setObjectName("pushButton\_14")  
 self.pushButton\_15 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_15.setGeometry(QtCore.QRect(109, 173, 75, 24))  
 self.pushButton\_15.setObjectName("pushButton\_15")  
 self.pushButton\_16 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_16.setGeometry(QtCore.QRect(22, 314, 75, 24))  
 self.pushButton\_16.setObjectName("pushButton\_16")  
 self.pushButton\_17 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_17.setGeometry(QtCore.QRect(109, 126, 75, 24))  
 self.pushButton\_17.setObjectName("pushButton\_17")  
 self.pushButton\_18 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_18.setGeometry(QtCore.QRect(109, 361, 75, 24))  
 self.pushButton\_18.setObjectName("pushButton\_18")  
 self.pushButton\_19 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_19.setGeometry(QtCore.QRect(109, 314, 75, 24))  
 self.pushButton\_19.setObjectName("pushButton\_19")  
 self.pushButton\_20 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_20.setGeometry(QtCore.QRect(109, 267, 75, 24))  
 self.pushButton\_20.setObjectName("pushButton\_20")  
 self.pushButton\_21 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_21.setGeometry(QtCore.QRect(196, 173, 75, 24))  
 self.pushButton\_21.setObjectName("pushButton\_21")  
 self.pushButton\_22 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_22.setGeometry(QtCore.QRect(283, 314, 75, 24))  
 self.pushButton\_22.setObjectName("pushButton\_22")  
 self.pushButton\_23 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_23.setGeometry(QtCore.QRect(196, 361, 75, 24))  
 self.pushButton\_23.setObjectName("pushButton\_23")  
 self.pushButton\_24 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_24.setGeometry(QtCore.QRect(370, 173, 75, 24))  
 self.pushButton\_24.setObjectName("pushButton\_24")  
 self.pushButton\_25 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_25.setGeometry(QtCore.QRect(196, 126, 75, 24))  
 self.pushButton\_25.setObjectName("pushButton\_25")  
 self.pushButton\_26 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_26.setGeometry(QtCore.QRect(370, 126, 75, 24))  
 self.pushButton\_26.setObjectName("pushButton\_26")  
 self.pushButton\_27 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_27.setGeometry(QtCore.QRect(196, 267, 75, 24))  
 self.pushButton\_27.setObjectName("pushButton\_27")  
 self.pushButton\_28 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_28.setGeometry(QtCore.QRect(196, 314, 75, 24))  
 self.pushButton\_28.setObjectName("pushButton\_28")  
 self.pushButton\_29 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_29.setGeometry(QtCore.QRect(370, 314, 75, 24))  
 self.pushButton\_29.setObjectName("pushButton\_29")  
 self.pushButton\_30 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_30.setGeometry(QtCore.QRect(283, 173, 75, 24))  
 self.pushButton\_30.setObjectName("pushButton\_30")  
 self.pushButton\_31 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_31.setGeometry(QtCore.QRect(283, 126, 75, 24))  
 self.pushButton\_31.setObjectName("pushButton\_31")  
 self.pushButton\_32 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_32.setGeometry(QtCore.QRect(283, 267, 75, 24))  
 self.pushButton\_32.setObjectName("pushButton\_32")  
 self.pushButton\_33 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_33.setGeometry(QtCore.QRect(370, 455, 75, 24))  
 self.pushButton\_33.setObjectName("pushButton\_33")  
 self.pushButton\_34 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_34.setGeometry(QtCore.QRect(196, 455, 75, 24))  
 self.pushButton\_34.setObjectName("pushButton\_34")  
 self.pushButton\_35 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_35.setGeometry(QtCore.QRect(283, 455, 75, 24))  
 self.pushButton\_35.setObjectName("pushButton\_35")  
 self.pushButton\_36 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_36.setGeometry(QtCore.QRect(109, 220, 75, 24))  
 self.pushButton\_36.setObjectName("pushButton\_36")  
 self.pushButton\_37 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_37.setGeometry(QtCore.QRect(22, 220, 75, 24))  
 self.pushButton\_37.setObjectName("pushButton\_37")  
 self.pushButton\_38 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_38.setGeometry(QtCore.QRect(283, 220, 75, 24))  
 self.pushButton\_38.setObjectName("pushButton\_38")  
 self.pushButton\_39 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_39.setGeometry(QtCore.QRect(370, 220, 75, 24))  
 self.pushButton\_39.setObjectName("pushButton\_39")  
 self.pushButton\_40 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_40.setGeometry(QtCore.QRect(196, 220, 75, 24))  
 self.pushButton\_40.setObjectName("pushButton\_40")  
 self.pushButton\_41 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_41.setGeometry(QtCore.QRect(370, 361, 75, 24))  
 self.pushButton\_41.setObjectName("pushButton\_41")  
  
 buttons = [  
 self.pushButton, self.pushButton\_2, self.pushButton\_3, self.pushButton\_4,  
 self.pushButton\_5, self.pushButton\_6, self.pushButton\_7, self.pushButton\_8,  
 self.pushButton\_9, self.pushButton\_10, self.pushButton\_11, self.pushButton\_13,  
 self.pushButton\_14, self.pushButton\_15, self.pushButton\_16, self.pushButton\_17,  
 self.pushButton\_18, self.pushButton\_19, self.pushButton\_20, self.pushButton\_21,  
 self.pushButton\_22, self.pushButton\_23, self.pushButton\_24, self.pushButton\_25,  
 self.pushButton\_26, self.pushButton\_27, self.pushButton\_28, self.pushButton\_29,  
 self.pushButton\_30, self.pushButton\_31, self.pushButton\_32, self.pushButton\_33,  
 self.pushButton\_34, self.pushButton\_35, self.pushButton\_36, self.pushButton\_37,  
 self.pushButton\_38, self.pushButton\_39, self.pushButton\_40, self.pushButton\_41  
 ]  
  
  
 for button in buttons:  
 button\_font = button.font()  
 button\_font.setPointSize(12)  
 button.setFont(button\_font)  
  
 self.retranslateUi(Form)  
 QtCore.QMetaObject.connectSlotsByName(Form)  
  
 def retranslateUi(self, Form):  
 \_translate = QtCore.QCoreApplication.translate  
 Form.setWindowTitle(\_translate("Form", "Calculator"))  
 self.pushButton.setText(\_translate("Form", "MS"))  
 self.pushButton\_2.setText(\_translate("Form", "cos"))  
 self.pushButton\_3.setText(\_translate("Form", "pi"))  
 self.pushButton\_4.setText(\_translate("Form", "^"))  
 self.pushButton\_5.setText(\_translate("Form", "7"))  
 self.pushButton\_6.setText(\_translate("Form", "4"))  
 self.pushButton\_7.setText(\_translate("Form", "1"))  
 self.pushButton\_8.setText(\_translate("Form", "="))  
 self.pushButton\_9.setText(\_translate("Form", "2"))  
 self.pushButton\_10.setText(\_translate("Form", "6"))  
 self.pushButton\_11.setText(\_translate("Form", "5"))  
 self.pushButton\_13.setText(\_translate("Form", "("))  
 self.pushButton\_14.setText(\_translate("Form", "C"))  
 self.pushButton\_15.setText(\_translate("Form", "sin"))  
 self.pushButton\_16.setText(\_translate("Form", "+"))  
 self.pushButton\_17.setText(\_translate("Form", "MR"))  
 self.pushButton\_18.setText(\_translate("Form", "8"))  
 self.pushButton\_19.setText(\_translate("Form", "-"))  
 self.pushButton\_20.setText(\_translate("Form", "e"))  
 self.pushButton\_21.setText(\_translate("Form", "tan"))  
 self.pushButton\_22.setText(\_translate("Form", "/"))  
 self.pushButton\_23.setText(\_translate("Form", "9"))  
 self.pushButton\_24.setText(\_translate("Form", "√"))  
 self.pushButton\_25.setText(\_translate("Form", "MC"))  
 self.pushButton\_26.setText(\_translate("Form", "M-"))  
 self.pushButton\_27.setText(\_translate("Form", "ln"))  
 self.pushButton\_28.setText(\_translate("Form", "\*"))  
 self.pushButton\_29.setText(\_translate("Form", "+/-"))  
 self.pushButton\_30.setText(\_translate("Form", "lg"))  
 self.pushButton\_31.setText(\_translate("Form", "M+"))  
 self.pushButton\_32.setText(\_translate("Form", "3√"))  
 self.pushButton\_33.setText(\_translate("Form", "."))  
 self.pushButton\_34.setText(\_translate("Form", "3"))  
 self.pushButton\_35.setText(\_translate("Form", "0"))  
 self.pushButton\_36.setText(\_translate("Form", "1/x"))  
 self.pushButton\_37.setText(\_translate("Form", "x!"))  
 self.pushButton\_38.setText(\_translate("Form", "x^2"))  
 self.pushButton\_39.setText(\_translate("Form", "x^3"))  
 self.pushButton\_40.setText(\_translate("Form", "10^x"))  
 self.pushButton\_41.setText(\_translate("Form", ")"))

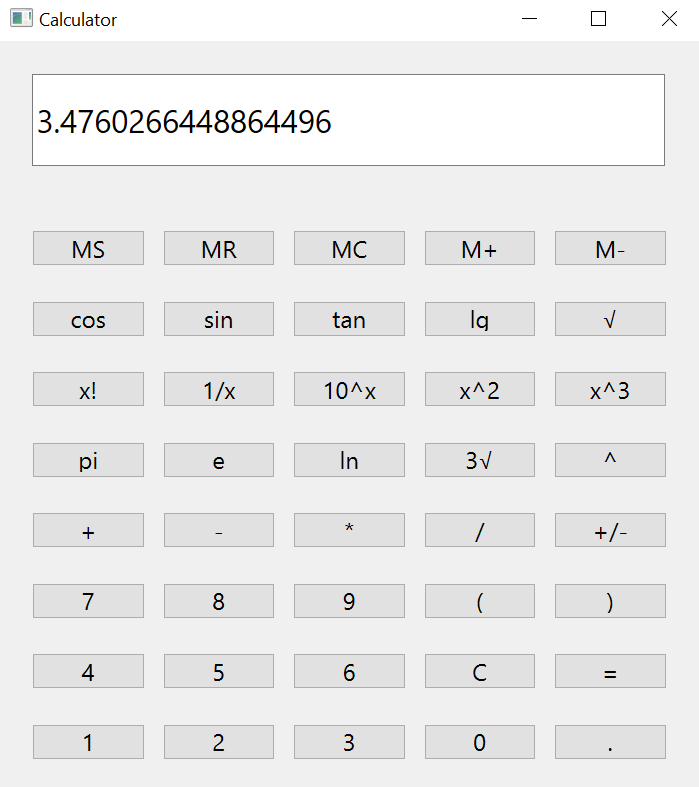
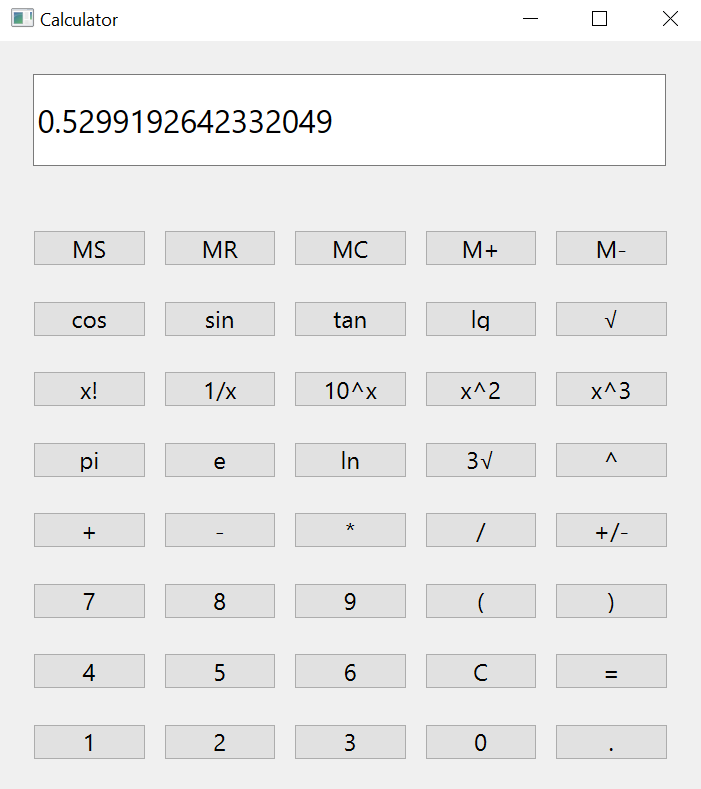
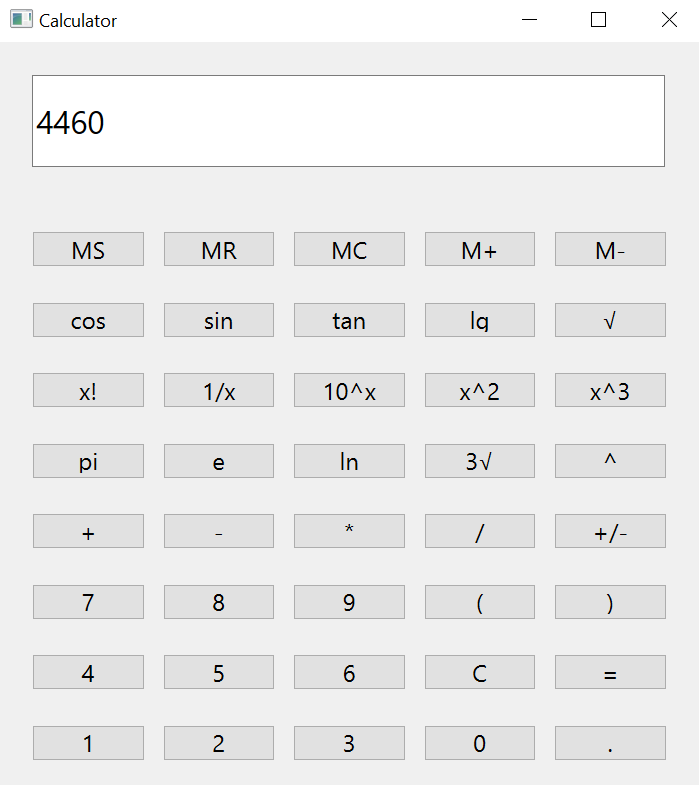
**Скрін-шоти виконання завдання лабораторної роботи :**

**Графічний інтерфейс програми :**

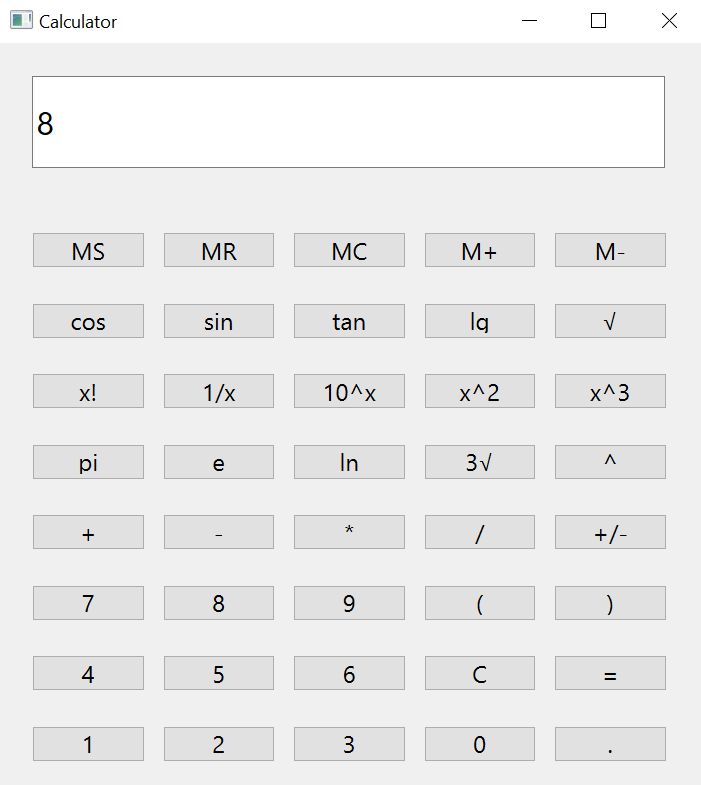
****

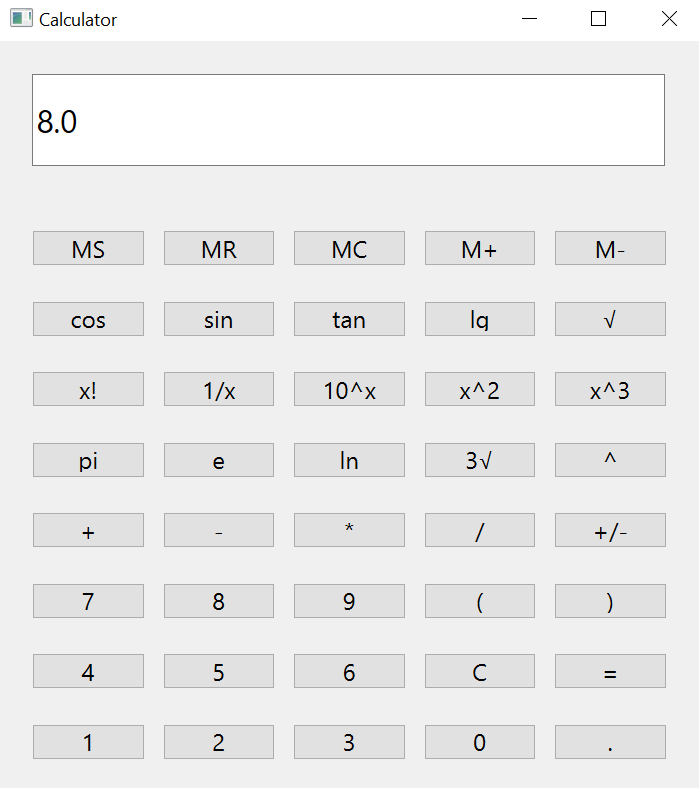
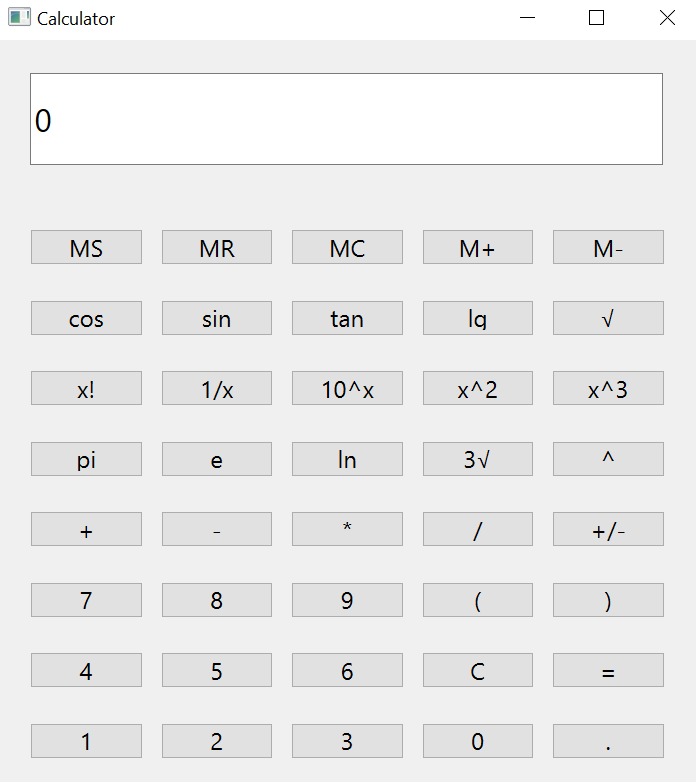
**Конторльні приклади для демонстрування роботи програми :**

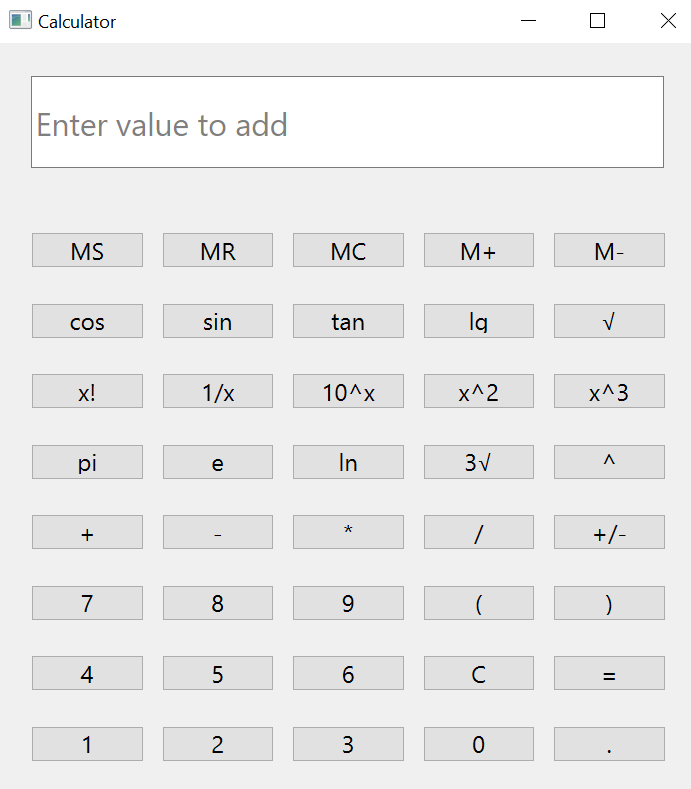
****

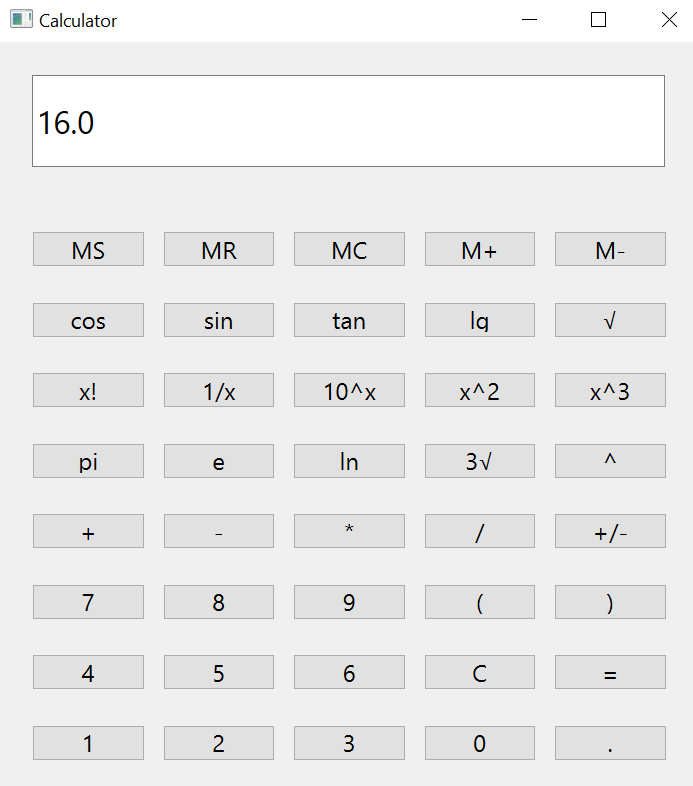
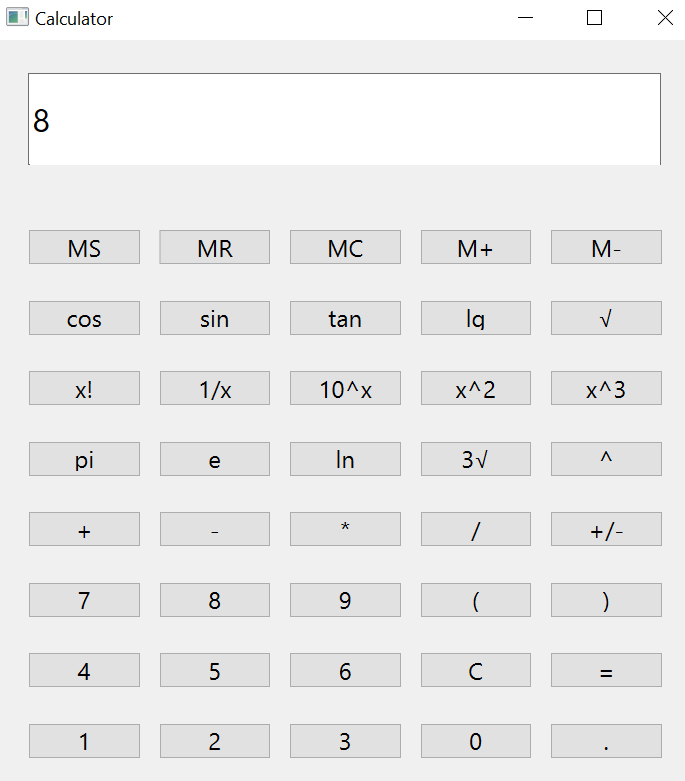
****

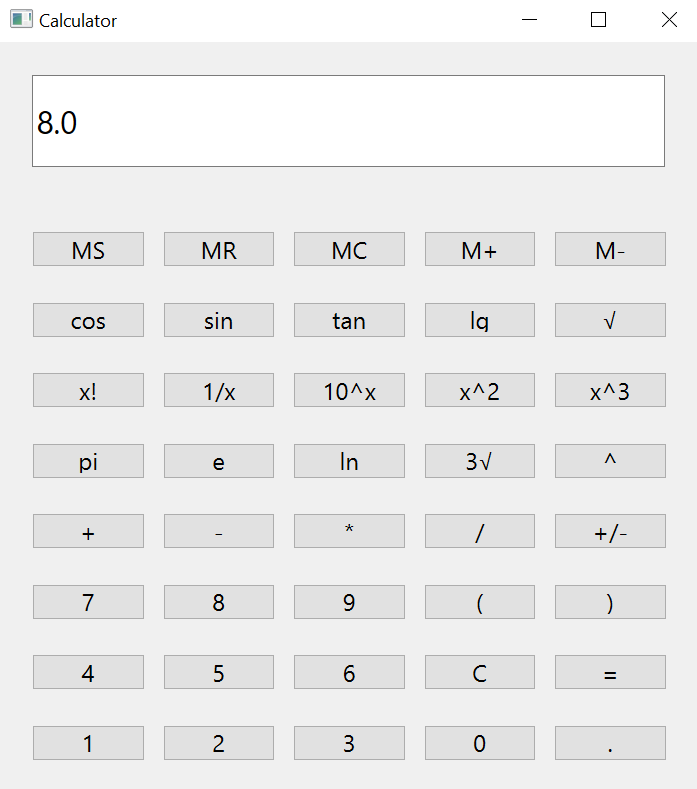
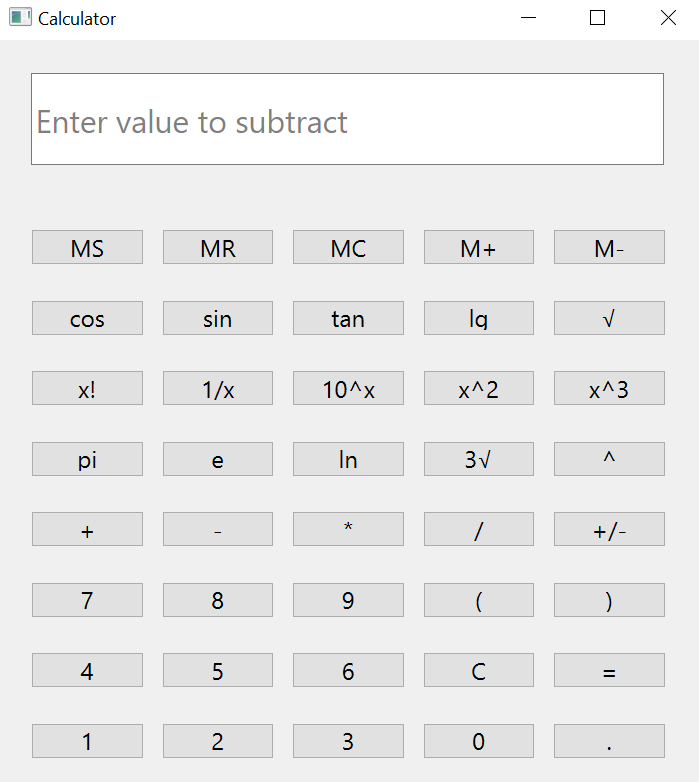
**Функції роботи з пам’яттю :**

1. **MS і MR :   
   **

****

1. **Функції M+ і M- :** 

****

****